CLEF Quick start Guide:

CLEF:

CLEF (CIRMMT Live Electronics Framework) is a comprehensive event based performance patch for live electronics. Using CLEF, composers can quickly create complex DSP, routing and playback events, relying on CLEF's messaging infrastructure and feature set.

Basic Concepts:

CLEF applies various signal processing and analysis to audio via modules which can be connected to each other and are controlled via messages on a shared message bus . This requires the construction of four inter-related elements. Modules, routings, events and cue list.

Modules:

Modules form the processing core of CLEF. They chain together, generating data and processing incoming audio and control messages. Modules are selected at the beginning of the process, and act as a pool of resources, each containing controllable parameters that alter its function and characteristics.

Routings:

A routing describes a specific topology of signal connections between modules, in other words the 'path' of audio signals through your live-electronics system.

Events:

Events are sets of instructions for the modules and router, not unlike a page of a score. Each event is composed of a series of instructions that include changes to: routings, module parameters. (Note, that events can function like 'presets', but they can also contain algorithms (in the form of Max patches))

Cue List:

The cues list manages the sequence of events that form the composition. Each cue is assigned to an event, which are triggered in a specific order. The cue list is flexible and allows for the sequence of events to be re-ordered, and events to be re-used.

Navigation:

Max mode vs. CLEF mode:

CLEF has two modes that can be accessed by clicking the dropdown menu in the top left of the performance window: Or, can be accessed via keyboard: [shift+<] and [shift+>]

CLEF Windows:

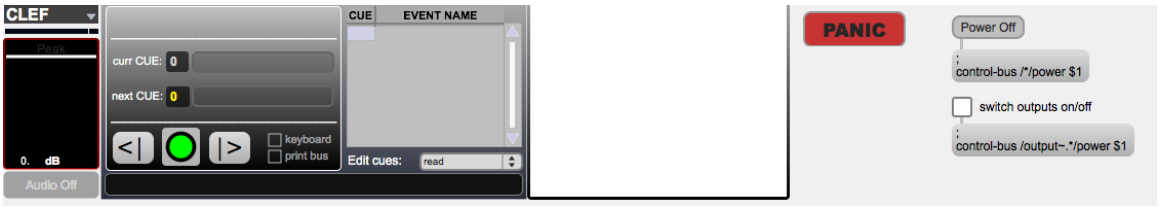
Clef has three main *Views* that can be accessed in clef mode via "Views" in the Max menu bar. They can be quickly accessed via shortcut.

Command+1: Performance View
Command+2: Modules View
Command+3: Events View

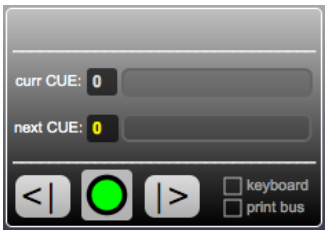
Pro Tip: Navigating clef is much more efficient with keyboard shortcuts. Use them!

Performance View: (Creating Cues)

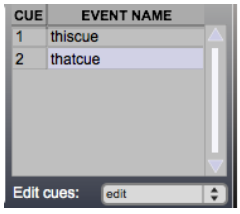
The performance window gives an overview of the current state of the patch, and the position within a composition. It can be accessed under Views➔Performance.



1. Audio On / Off: This section toggles audio on and off as well as showing incoming and outgoing audio signals.
2. CUE Navigator: This section navigates though cues and displays events. Selecting and triggering event cues is done from this section.



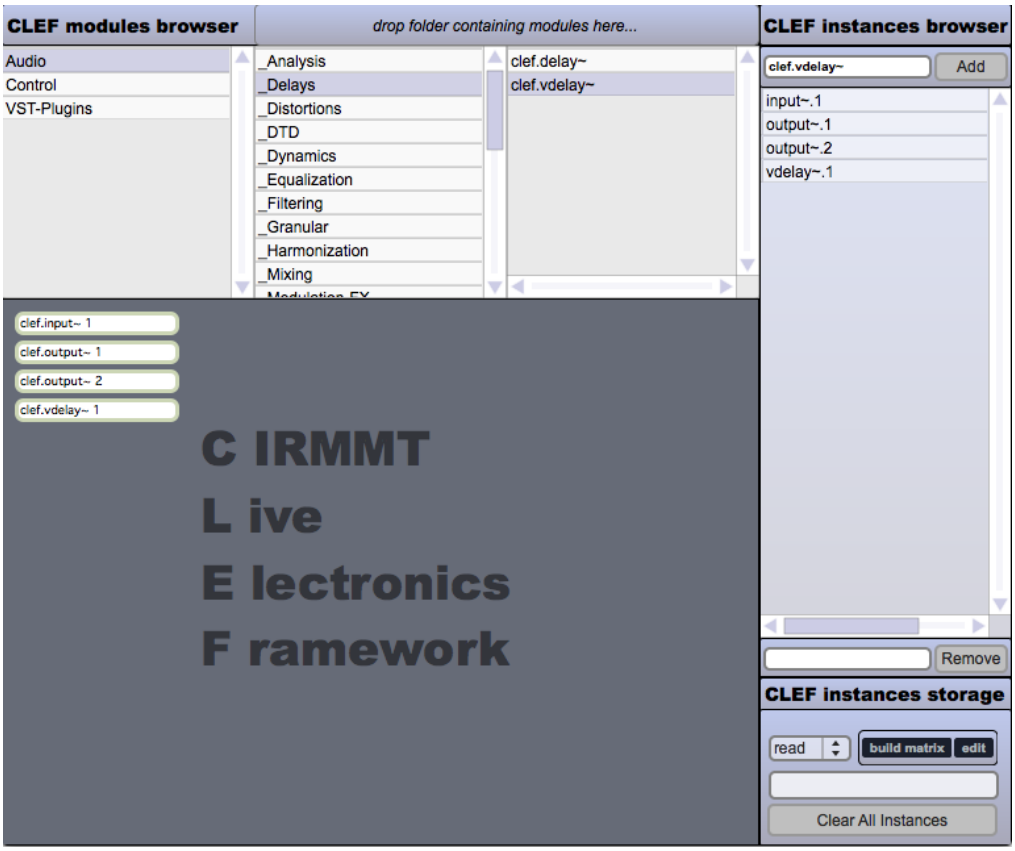
3. Cue List: This section creates and orders event cues. The cue list is stored in a [coll] object. Different actions can be accessed via the dropdown menu ("edit" for example, gives direct access to the coll-file.) The cues visible in the GUI correspond to "events" that are created in the event section.



Pro Tip: Events and cues are different! An event is a set of instructions that will change the stage of the patch. Events can vary over time. Cues trigger events, and can be re-ordered in the cue list at any time.

Pro Tip: It is important to make sure that the names of your cues correspond exactly to names of events in the event window. (See: Events)

Modules View: (instantiating dsp-modules)



1. **Access the Modules window** under Views➔Modules. Or, with the keyboard shortcut [command+2] (Clef Mode)

2. **Adding Modules:** Navigate the module browser and select desired modules. Click "add" in the clef instances browser and the module will be instantiated. (You can also dbl-click on the entry in the browser). Once modules are instantiated you will see an entry added in the list on the right (instances browser). In addition you will see an abstraction being instantiated on the bottom left window.

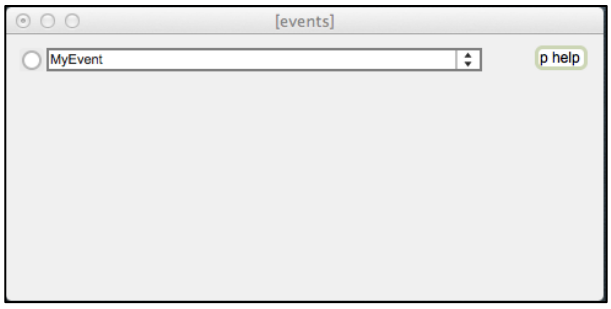
3. **Removing Modules:** Select the module from the CLEF instances browser and click [remove] at the bottom right of the window.

> **Pro Tip: Output modules must be created manually. In the modules browser: Audio➔Mixing➔clef.output~ Two clef.output~ for stereo, more for multichannel. The same is true for audio inputs.**

4. **Instance Storage:** Bottom right of the module browser contains a time saving option for saving and loading lists of modules.

> **Pro Tip: Once you're done instantiating modules you need to 'build' the dsp-matrix to which the modules are all connected. Click on the button "build matrix" on the bottom right.**

Events View: (Creating and managing events)



1. **Access the Events window** under Views➔events. Or, with the keyboard shortcut [command+3] (Clef Mode)

2. **Create new events:** New in CLEF v0.3, events and widgets can be pasted directly from the menu! Unlock the event window patch and*: ctrl+click → paste from → CLEF → wgt.event* to add events to the events window.

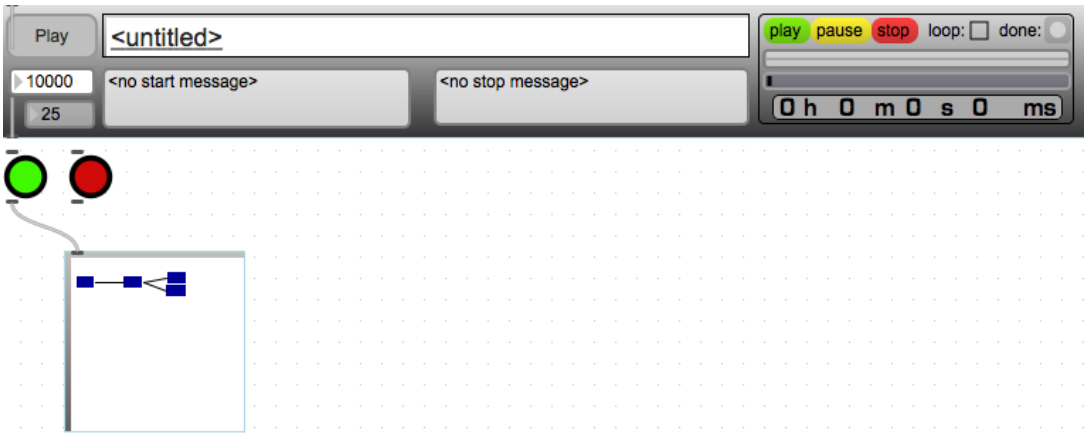
   Pro Tip: Make sure your event names correspond to your cue list. However, the ordering in the event window is not related to the order of the events as cued by the cue list.

3. **Edit events:** click the event object in the events window and select "Open". This panel is where events are constructed and routing / control data is managed.

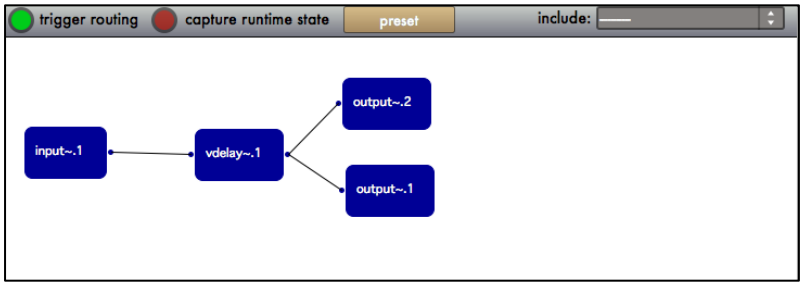
4. **Event Parameters:** Each event has a beginning, end, duration, start/stop message and name. These parameters can be entered in the event inspector's *event-top-pannel*. This event information will be displayed in the performance window to help keep track of event as the piece progresses though the cue list.

5. **Triggers:** When the cues inspector triggers an event, a bang is sent from the green button object within an event. A stop message will occur from the red button object after a set duration. By connecting these objects to message boxes, widgets and other max objects, the routing and DSP parameters are triggered and altered as the piece progresses.

Routing Window: (Create a Signal Routing)



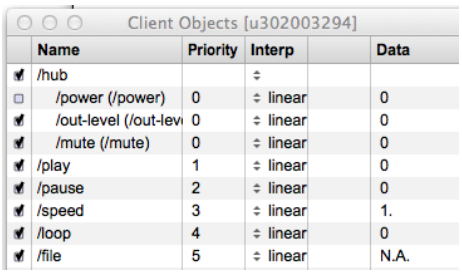
1. **Routings:** Routing presets change CLEF's module routing from within an event. When triggered, the routing widget updates the global routing presets, re-routing the signals between modules. Create a new routing (in Max mode) via: *ctrl+click* → *paste from* → *CLEF* → *wgt.routing* Double click the *routing miniview* to access the routing window below.



2. Any modules added to CLEF via the module window will show up in the *include* menu in the top right. Add modules by selecting them from the list. Use the GUI to connect the modules as required. The changes made here will be reflected in the routing mini view.

Pro Tip: Standard Max order of operations applies in the events window. Make sure your routing preset is updated before adjusting the parameters!

3. **Accessing Module Parameters:** From within the module widget, Double click a module to access a list of available parameters (a *pattr clientwindow)*. You can double-click into a field in the "Data" column to enter a value for a module parameter. This is helpful when creating presets on the fly.



4. **Gain:** Double click patch cords to access a gainstage feature.

Pro Tip: Each module has standard and non-standard settings as part of their namespace. "Power" needs to be activated for the module to function. If there is no sound, check the audio toggle in the performance window.

5. **Access the Connections window** When running a patch, check the current routing state via: *Views* ➔ *connections*. Or, with the keyboard shortcut *(Clef Mode) [command+5]*

Event Details:

When creating events in clef, there are useful resources to help manage, construct and control messages. These objects are available via option click (edit mode in max mode).

Option click → Paste From → CLEF → *select object from list*

Pro Tip: Using the 'paste from' feature will speed up event creation workflow a considerable amount.

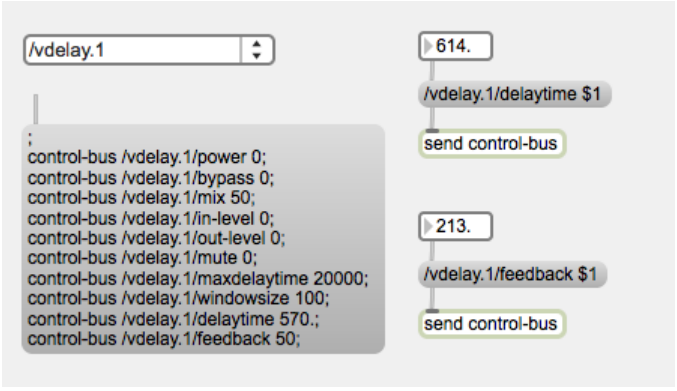
The following is a list of the most important, time saving objects included in CLEF.

Get module States: [clef.getmodulestate.wgt]

To quickly access module parameters use:

[bpatcher clef.getmodulestate.wgt]

(Max Mode) Option click → Paste From → CLEF → wgt.getmodulestate



Connect the outlet to the left inlet of a message box. Use the dropdown menu to select a module. The resulting information (message box) can be triggered as a module preset, or specific lines may be copied into other objects to send osc-messages for changing parameters.

Pro Tip: [bpatcher clef.getmodulestate.wgt] must be created AFTER the modules are instantiated. If the dropdown box is empty, re-create getmodulestate.wgt.

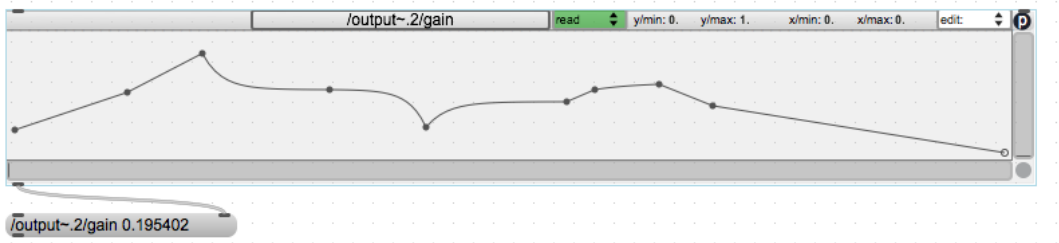
Changing The Routing of audio: [clef.routing.wgt]

Remember: CLEF's audio routing is updated within events using [clef.routing.wgt]. Use this widget, then connect it to a green button.

Pro Tip: Use the trigger object to control the order of execution, i.e. make sure your routing presets are changed before sending audio.

Control over time: [clef.osc-lane]

[bpatcher clef.osc-lane.absolute.wgt] and **[bpatcher clef.osc-lane.relative.wgt]** are useful objects for controlling module parameters over time. It automatically scales to the duration of the event, and allows designing, copying, rescaling, etc. of break point functions. For detailed information consult the help file.



Pro Tip: Double clicking on the box to the left of the green read/write dropdown opens a window which allows you to choose an available module parameter to control using menus.

Access these widgets via: *(Max Mode) Option click → Paste From → CLEF → wgt.osc-lane.obsolute / wgt.osc-lane.relative*

Absolute vs. Relative: [clef.osc-lane.absolute.wgt] preserves the scale of a user- defined break point function when the min and max values are adjusted.
[clef.osc-lane.relative.wgt] break point functions will be rescaled when changing min/max values of the lane.

Saving and Loading Clef Projects:

A clef project has three parts. Patch, Instances, Cues. Each stores a different aspect of the patch. The patch saves your events and the GUI objects you instantiated. The "Instances" file saves the modules that were instantiated in a project. 'Cues' saves the list of cues describing a sequence to events.

To save a project (in clef mode) select: Project → Save As…
You will be prompted to give file name / location of the cues and instances.

To open a project (in clef mode) select: Project → Open…
Follow the instructions.

In addition, you can always save Cues or Instances individually.
Note, that the patch needs to be saved manually (as in 'traditional' Max patching).

Pro Tip: It's a good practice to start a project with the CLEF-main.maxpat patcher, and then save it under another name (the project).

Additional Resources:

Getting Started:

You have already found the getting started guide, so you know where to find it. Just in case you forget:

CLEF/Documentation/gettingstarted.pdf

Help files:

Along with regular max help objects, most CLEF specific abstractions and widgets have their own help files:

Option click → open: clef.object.wgt help

Templates and Example Projects:

Exploring an already functioning patch is often the quickest way to understand the functionality of an object, or a new feature you didn't know existed. See:

CLEF/examples

And

CLEF/templates

Pro Tip: Example projects consist of a patcher file (.maxpat), and 3 textfiles (project, cues and instances). First open the .maxpat files, then load the project using the steps outlined on page 2 of the CLEF quick start guide.