

Contents

1	About	3
1.1	Features	3
1.2	Project Status	3
2	Getting Started	4
2.1	Cues	4
2.2	Layouts	4
2.3	Menus	4
2.4	Plugins	4
3	Application Menu	5
3.1	File menu	5
3.2	Edit menu	5
3.3	Layout menu	5
3.4	Tools	6
3.5	About	6
4	Cart Layout	7
4.1	Layout Operations	7
4.2	Layout Options	8
5	List Layout	10
5.1	User Interface	10
5.2	Layout Commands	11
5.3	Layout Options	11
6	Cues	13
6.1	Cue options	13
6.2	Media Cues	14
6.3	Action Cues	15
6.4	Editing multiple cues	17
7	GStreamer Backend - Media Settings	18
7.1	Input elements	19
7.2	Plugins elements	20
7.3	Output elements	21
8	GStreamer Backend - Custom Elements	23
8.1	Element Syntax	23
9	Command Cues Examples	24
9.1	LibreOffice	24
9.2	VLC	24
9.3	MPV PLAYER	24
10	ArtNet Timecode	25
10.1	How to use	25
10.2	Timecode Preferences	25
10.3	Timecode Cue Settings	26
11	Cue Controls	28
11.1	How to use	28
11.2	Keyboard	28

11.3 MIDI	29
12 Presets	31
12.1 How to use	31
13 ReplayGain & Normalization	33
13.1 How to use	33
14 Session URI Editor	34
14.1 How to use	34
15 Synchronization	35
15.1 How to use	35
15.2 How it works	35
16 Triggers	36
16.1 How to use	36
16.2 Events	36

1 About

Linux Show Player (or LiSP for short) is a free cue player designed for sound-playback in stage productions. The goal of the project is to provide a complete playback software for musical plays, theater shows and similar.

1.1 Features

Here a list of the main functionality offered by LiSP:

- Cart layout (buttons matrix) suited for touchscreens
- List layout suited for keyboards
- Large media-format support thanks to GStreamer
- Realtime sound effects: equalization, pitch shift, speed control, compression, ...
- Peak and ReplayGain normalization
- Undo/Redo changes
- Remote control over network, between two or more sessions
- ArtNet Timecode (via [OLA](#))
- MIDI support for cue triggering
- MIDI cues (send MIDI messages)
- Multi-language support (see [transifex](#) for a list of supported languages)

1.2 Project Status

Currently only GNU/Linux systems are supported.

The application is quite stable, is already been used for multiple performances by different people, but, due to the heterogeneous nature of the GNU/Linux ecosystem, my suggestion is to test it in a “working” environment to detect possible problems with some configuration.

2 Getting Started

Before diving into the different aspects of LiSP you need to understand the main concepts that set the basis on how the application works.

2.1 Cues

First, and probably the most important components you will find in LiSP, are the **cues**, a cue is used to execute a specific action in a repeatable manner, every cue allow customization of its behaviours independently.

Cues are at the heart of every show, allowing to play sounds, send MIDI messages, controls other cues and so on.

2.2 Layouts

When creating a new show in LiSP you have the ability to chose a *layout*, this will affect how cues will be displayed and eventually provides different sets of features.

Currently two layouts are provided:

- **List Layout:** arrange the cues in a list and provided a mouse/keyboard oriented UI
- **Cart Layout:** arrange the cues as buttons in one or more grids and provide a more touch-oriented UI.

2.3 Menus

Most of the functionality are accessible via the top-bar menu, here a small explanation on what you will find:

- **File:** Operation related to the current session or the global application
- **Edit:** Functions mainly related to adding/editing cues (accessible right-clicking on empty areas of the layout)
- **Layout:** Functions provided by the current layout
- **Tools:** Utility to make life easier

Cues provides a contextual (right-click) menu to access cue-specific options.

2.4 Plugins

Linux Show Player is heavily based on plugins, while this is almost always hidden from the user, most of the functionality are provided via plugins. From time to time the documentation may refer to those as plugins or modules.

3 Application Menu

Here you can find a comprehensive list of the standard main menu entries. Keyboard shortcuts are marked like this [key-combination].

Note:

- Some of the shortcuts may be different based on the used locale (language)
 - In *Ubuntu Unity (7 and earlier)* shortcuts will not work
-

3.1 File menu

- **New session:** [CTRL+N] Open a new (empty) session, if the current session is not saved a confirmation dialog is shown.
- **Open:** [CTRL+O] Open a saved session, if the current session is not saved a confirmation dialog is shown.
- **Save session:** [CTRL+S] Save the current session.
- **Save with name:** [CTRL+SHIFT+S] Save the current session in a new file.
- **Preferences:** Open the application settings dialog.
- **Toggle fullscreen:** [F11] Enable/Disable fullscreen mode.
- **Exit:** Close “Linux Show Player”, if the current session is not saved a confirmation dialog is shown.

3.2 Edit menu

- **Action cues:** Allow to add different types of action-cues.
- **Media cues:** Allow to add media-cues.
- **Media cue (from file):** [CTRL+M] Allow to add multiple (audio) cues selecting the multimedia files.
- **Undo:** [CTRL+Z] Undo the last action (the last action is shown in the bottom of the window).
- **Redo:** [CTRL+Y] Redo the last undone action.
- **Select all:** [CTRL+A] Select all the cues.
- **Select all media cues:** Select all and only the media-cues
- **Deselect all:** [CTRL+SHIFT+A] Deselect all the cues
- **Invert selection:** [CTRL+I] Invert the selection of all the cues.
- **Edit selected cues:** [CTRL+SHIFT+E] Open a multi-edit dialog for the selected cues.

3.3 Layout menu

This menu gives access to layout functionality and display options for the current view.

3.3.1 Cart Layout

- **Add page/Add pages:** Allows to add one or multiple pages on the current layout. Pages can be switched using the tab bar on top of the layout or directional keys.
- **Remove current page:** Remove the current page and all its cues
- **Countdown mode:** Cues will display the remaining time instead of the elapsed time.
- **Show seek-bars:** Media cues will display a seek bar, allowing to directly seek to a specific time of the cue.
- **Show dB-meters:** Media cues will display a dB meter on their right side.
- **Show volume:** Media cues will display a volume control on their right side. Setting the volume to the middle point (50%) of the slider sets the volume to +0dB.
- **Show accurate time:** When checked, cues will display play time with a precision of 0.1s. When unchecked the time is only precise down to 1s.

3.3.2 List Layout

- **Show playing cues:** Show / hide the “Right List” and list control buttons;
- **Show dB-Meters:** Show / hide the db-meters for running media-cues;
- **Show seek-bars:** Show / hide seek bars for running media-cues;
- **Show accurate time:** Show / hide tens of seconds for running media-cues;
- **Auto-select next cue:** If enabled the next cue is selected automatically.

3.4 Tools

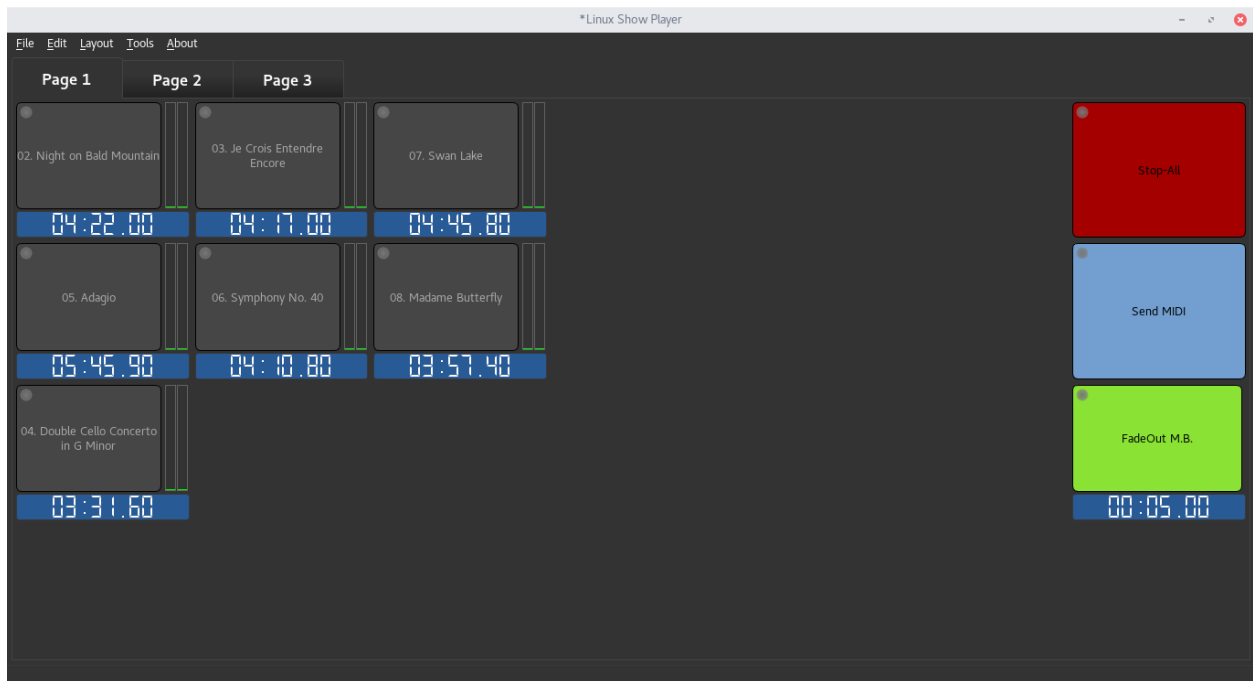
Plugins can add their entries or sub-menus here (see the plugins pages).

3.5 About

Info about “Linux Show Player” and some of the used technologies.

4 Cart Layout

The Cart Layout organize all the cues in grid-like tabs, cues are shown as buttons, as in the following image:



If the cue provides a duration, the current cue time is shown at the bottom of the button.

4.1 Layout Operations

4.1.1 Adding/Removing Pages

Pages will be added automatically when needed (this behavior can be disabled), or manually, to do so two options are provided, in the top-bar `Layout > Add page` and `Layout > Add pages`, the first will add a single page (at the end), the second will show a dialog that allow to insert a custom number of pages.

To remove a page, select the page to be removed, then `Layout > Remove current page`, a confirmation dialog will be shown, if `Yes` is clicked, then the page (and cues) will be deleted.

4.1.2 Change page

Pages can be switched using the tab bar on top of the layout or directional keys.

4.1.3 Cues Execution

A cue can be start/stopped simply `Left-Clicking` on it.

4.1.4 Cues Editing

The setting dialog for a cue can be opened in two ways: `Right-Click > Edit cue` or `SHIFT+Right-Click`.

Cues can be selected/deselected for multi-editing with `Right-Click > Select` or `CTRL+Left-Click`.

4.1.5 Move and Copy Cues

Cues can be copied or moved (into free spaces) inside a page or between different pages:

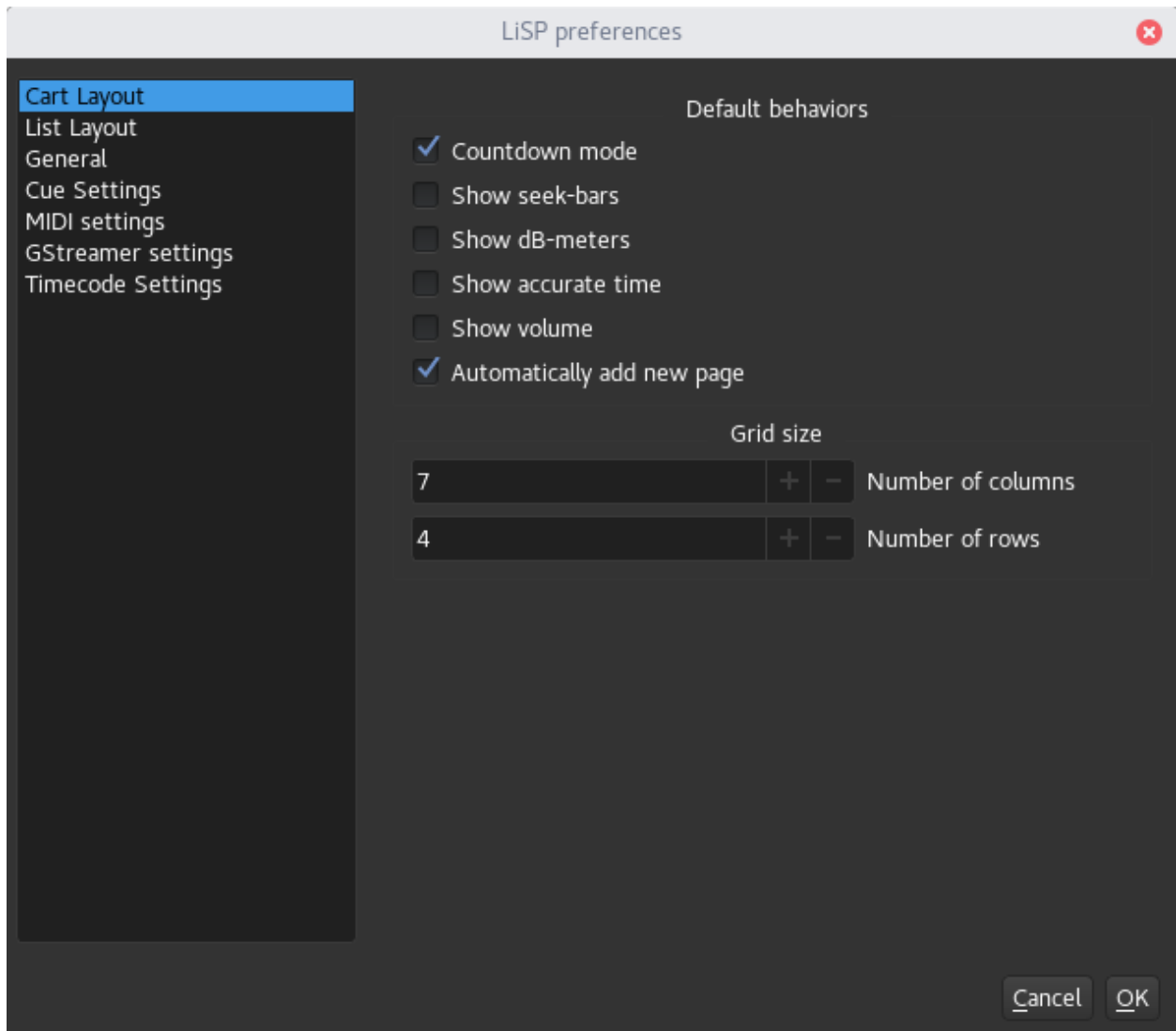
- **Move:** cues can be moved with CTRL+Drag&Drop
- **Copy:** cues can be copied with SHIFT+Drag&Drop

4.2 Layout Options

In the application settings (File > Preferences) various options are provided:

- **Countdown mode:** when enabled the current cue time is displayed as a countdown
- **Show seek-bars:** when enabled a slider able to change the current playing position of media cues (for media cues)
- **Show dB-meters:** when enabled, a db level indicator is shown (for media-cues)
- **Show accurate time:** when enabled the cue time is displayed including tens of seconds
- **Show volume:** when enabled a volume slider is shown (for media-cues)
- **Automatically add new page:** when enabled new pages will be created when the current pages are full and new cues are added
- **Grid size:** define the number of rows & columns per page. (require to reload the session)

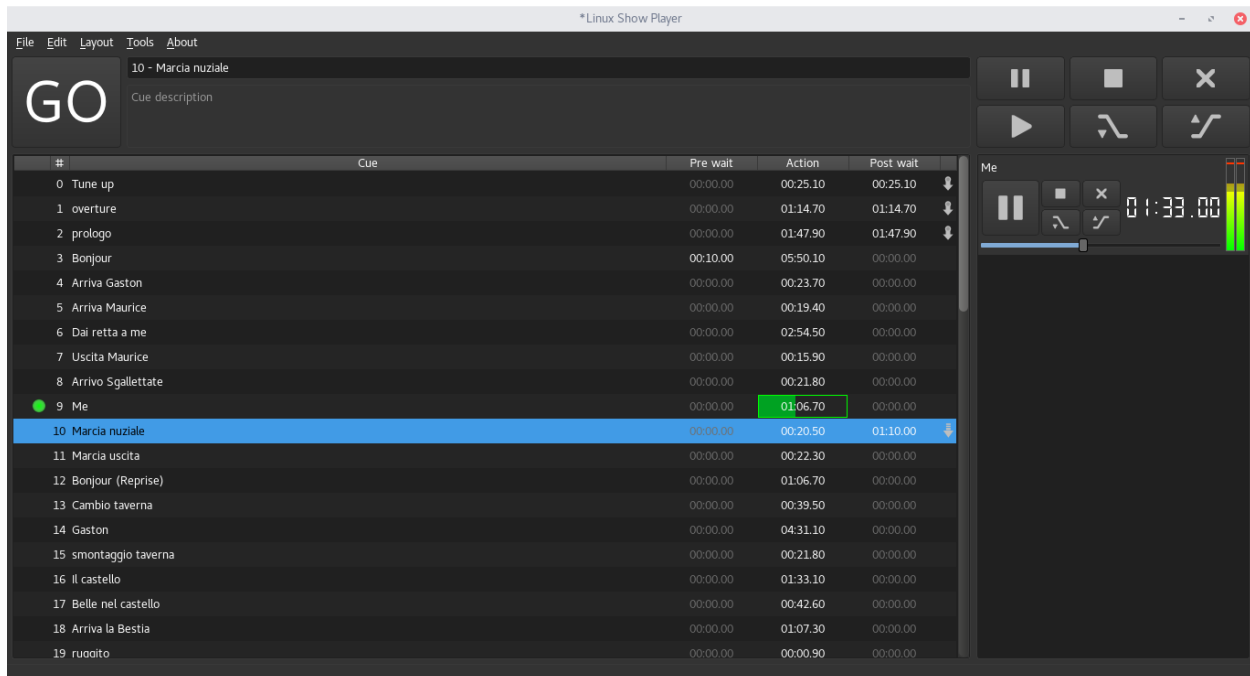
Warning: When the grid size is changed, cues will be visually shifted to keep their logical positioning.



Note: Cart Layout does not support cues “next-action”.

5 List Layout

The List Layout, as the name suggest, organize the cues in a (single) list, a sidebar to monitor and interact with the running cues is also provided.



5.1 User Interface

5.1.1 Top Panel

- **Top left:** we can find the GO button, this will execute the selected cue and go forward;
- **Top center:** name and description of the current cue are displayed here;
- **Top right:** list-control commands are provided, those allow to stop, pause, restart, interrupt and fade all the cues in the list.

5.1.2 Left List

All the cues are shown here in a list-like view, the following column are shown:

- **(status):** the first column show the current state of the cue running/paused/error and a selection indicator
- **#:** The cue index
- **Cue:** The cue name
- **Pre wait:** Pre wait indicator
- **Action:** Cue time indicator
- **Post wait:** Post wait indicator
- **(next action):** What should be done after “post wait”

5.1.3 Right (running) List

The running cues are shown here, you can stop, pause/restart, interrupt and fade single cues.

5.2 Layout Commands

5.2.1 Navigate the cues

To change the current cue, directional keys can be used to go up and down into the list, alternatively Left-Clicking the cue will set it as current.

5.2.2 Cues Execution

To execute the current cue, press Space, (can be changed in the layout options) or use the GO button.

5.2.3 Cues Editing

The setting dialog for a cue can be opened in two ways: Right-Click > Edit cue or Double-Click the cue.

Cues can be selected/deselected with Right-Click > Select, CTRL+Space or CTRL+Click

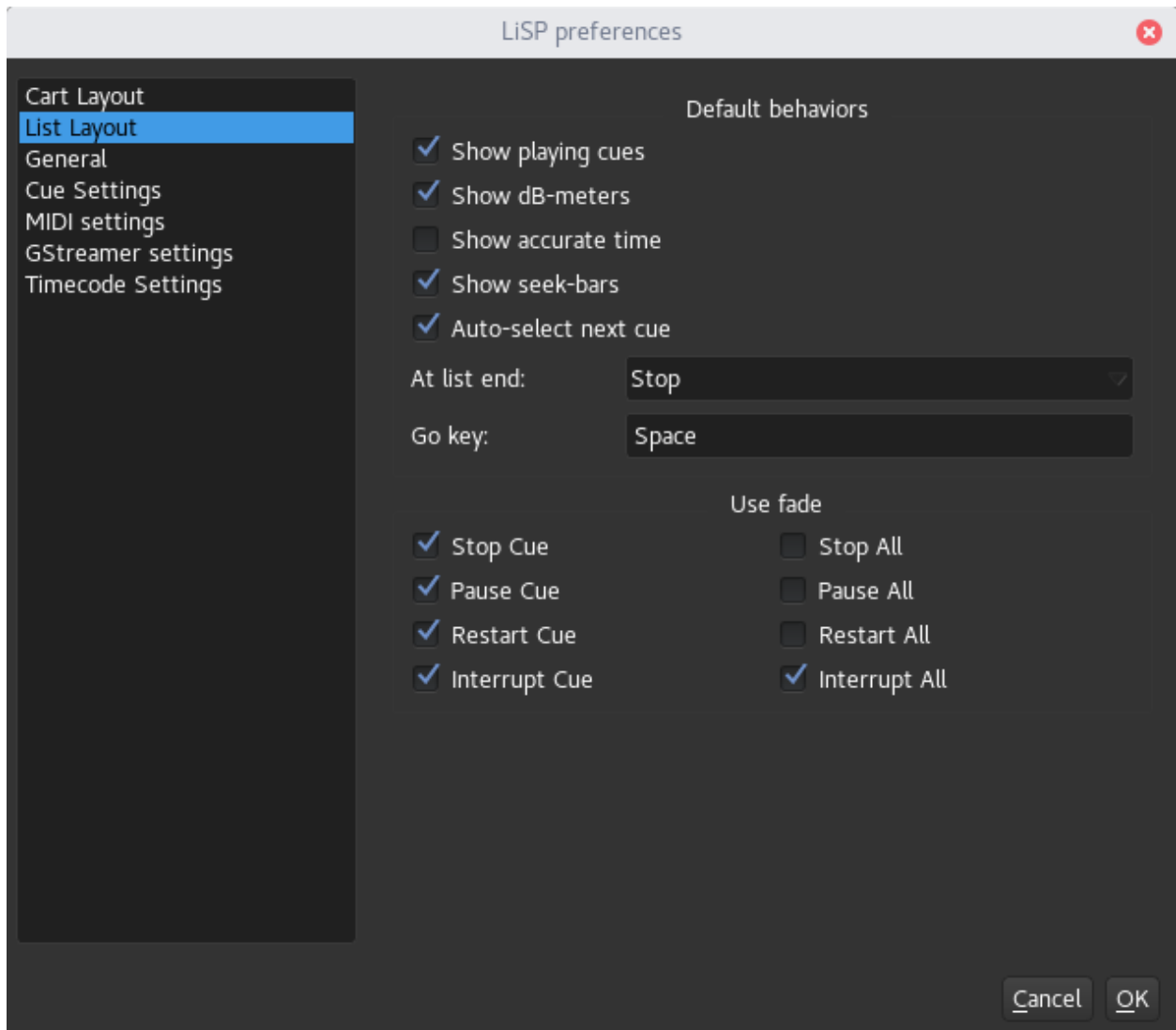
5.2.4 Move and Copy Cues

- **Move:** cues can be moved with a simple Drag&Drop
- **Copy:** cues can be copied using CTRL+Drag&Drop

5.3 Layout Options

In the application settings (File > Preferences) various options are provided:

- **Show playing cues:** show/hide the “Right List” and list control buttons
- **Show dB-Meters:** show/hide the db-meters for running media-cues
- **Show accurate time:** show/hide tens of seconds for running media-cues
- **Show seek-bars:** show/hide seek bars for running media-cues
- **Auto-select next cue:** if disabled the next cue will not be selected automatically
- **At list end:**
 - **Stop:** the selection doesn’t change when the last cue is executed
 - **Restart:** the selection is moved back to the first cue
- **Go key:** define up to 4 key-combinations that can be used to execute the current cue, to do so, double click the edit-area, then enter your keys combinations
- **Use fades:** when disabled the corresponding buttons on the right-panel executes their action without fades.



6 Cues

Cues are the main component of every show/session. There are multiple types of cues able to perform different tasks, those can be subdivided in two main categories:

- **Media cues:** used to play multimedia contents, usually related to some media file or stream
- **Action cues:** used to accomplish more complex interaction (e.g. start multiple cues at one), fading other cues parameters or interact with external devices or application.

A cue can perform different *actions* depending on its current *state*

Actions:

- **start:** Perform the cue task
- **stop:** Stop the running cue
- **pause:** Pause the running cue if possible
- **interrupt:** Stop the running cue, other cues/functions will ignore this event
- **fade:** Decrease/Increase gradually a predefined cue parameter (e.g. volume)

Every action (except for fading) can be performed with or without a fadein/out.

6.1 Cue options

Cue options can be edited via a dialog, the way to access this dialog is described in the layouts pages. Options are organized in tabs depending on their context.

Two tabs are provided for all the cues (excluding plugins):

6.1.1 Appearance

Visual options (*some of them can be ignored by the layout*)

- **Cue name:** The name that identifies the cue
- **Description/Note:** A text for writing notes about the cue
- **Font size:** The font used to display the name
- **Font color:** The color of the font used to display the name
- **Background color:** The background color of the cue

6.1.2 Cue

General options for the cue, organized in 3 sub-tabs

Behaviors

Define the default actions used by the cue, this allows to disable fades by default, or to pause instead of stopping.

- **Start:** Action used to start the cue
- **Stop:** Action used to stop the cue

Pre/Post Wait

- **Pre wait:** Add a delay before the cue is started
- **Post wait:** Delay before `Next action` is executed
- **Next action: What to do after Post wait (can be ignored by the layout)**
 - *Do Nothing:* You know ...
 - *Auto Next:* Execute the next cue
 - *Auto Follow:* When the cue end, execute the next cue (`Post wait` value is ignored)

Fade In/Out

- **Fade In: Gradually increase a predefined value on faded(in)-actions**
 - **Duration:** How long the fade should last before reaching a maximum value
 - **Curve:** How the value should increase in time
 - **Fade Out: Gradually decrease a predefined value on faded(out)-actions**
 - **Duration:** How long the fade should last before reaching a minimum value
 - **Curve:** How the value should decrease in time
-

6.2 Media Cues

6.2.1 Audio Cues

Audio cues allow you to playback audio files.

A media cue doesn't playback directly the media, but rely on other components, those can be configured adding and removing effects or controls (e.g. volume, equalizer, speed).

Options

- **Media-Cue**
 - **Start time:** Time to skip from the media beginning
 - **Stop time:** Time to skip before the media end
 - **Loop:** Number of repetitions after first play (-1 is infinite)
- **Media Settings: Media options provided by the backend**
 - *GStreamer backend*

Video playback support is already planed but not yet implemented

6.3 Action Cues

6.3.1 Collection Cue

As the name suggest this cue allow to execute multiple cues at once, for each cue a different action can be specified.

Options (Edit Collection)

You can Add/Remove cues to the collection via the provided buttons, `Double-Click` values to edit them.

6.3.2 Stop All

This cue simply stop all the running cues, alternately can be configured to execute different actions.

6.3.3 Seek Action

This cue allow to seek a media cue to a specified point.

Options (Seek Settings)

- **Cue:** The target media-cue (a button is provided to select the target)
 - **Seek:** The time-point to reach
-

6.3.4 Volume Control

A Volume Control cue allows to trigger a volume change/fade-in/out on a selected media cue.

Options (Volume Settings)

- **Cue:** The target media-cue (a button is provided to select the target)
 - **Volume:** The volume to reach
 - **Fade: Volume fading options**
 - **Duration:** The volume fade duration in duration (if 0 the change is instantaneous)
 - **Curve:** The fade curve
-

6.3.5 MIDI Cue

A MIDI cue allow to send a MIDI message to the MIDI output device used by the application (can be selected in the application preferences).

Options (MIDI Settings)

- **MIDI Message:** Set what type of message to send
- **(message attributes):** Depending on the message type different attribute can be edited

Supported MIDI messages

- note*on
 - note*off
 - control*change
 - program*change
 - polytouch
 - pitchwheel
 - song*select
 - songpos
 - start
 - stop
 - continue
-

6.3.6 Command Cue

This cue allow to execute a shell command, until the command runs the cue is `running` and can be stopped, doing so will terminate the command.

To see the command output, LiSP should be launched from a terminal, and `Discard command output` must be disabled.

Options (Command Cue)

- **Command:** the command line to be executed (as in a shell)
- **Discard output:** when enabled the command output is discarded
- **Ignore command errors:** when enabled errors are not reported
- **Kill instead of terminate:** when enable, on stop, the command is killed (abruptly interrupted by the OS)

For examples of commands to control external programs, see [here](#).

6.3.7 Index Action

This cue give the ability to execute a specific action on a cue in a given position in the layout.

Options (Action Settings)

- **Index**
 - **Use a relative index:** When toggled the position is considered relative to the current cue position.
 - **Target index:** The position of the target (the UI will enforce a valid index)
 - **Action:** The action to execute
-

6.4 Editing multiple cues

You can select all cues at once using `Edit > Select All (CTRL+A)`, while multiple cues are selected, you can use `Edit > Edit selected media [CTRL+SHIFT+E]`, to edit multiple cues at once.

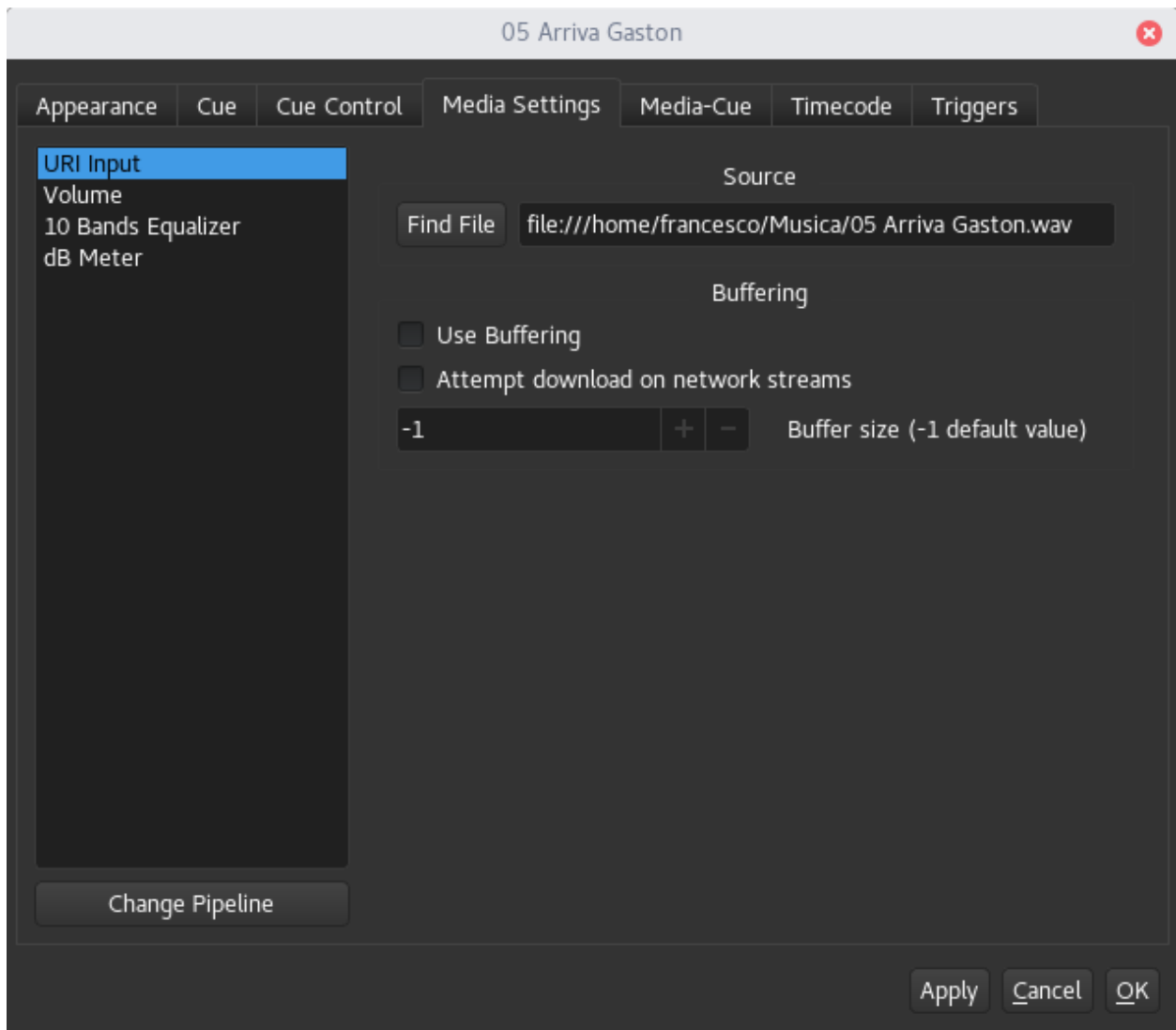
The available options will depend on the types of the selected cues.

7 GStreamer Backend - Media Settings

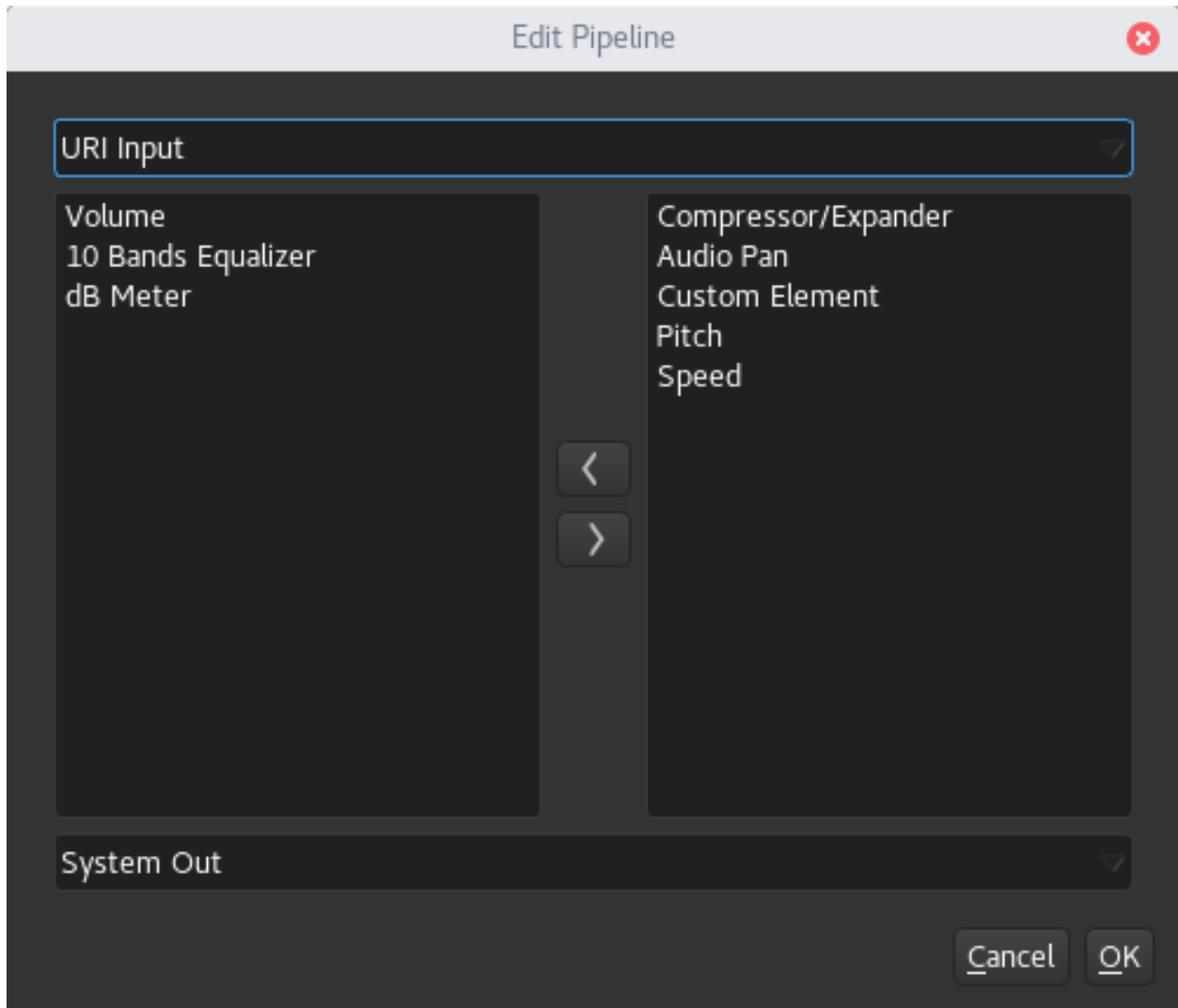
Media Cues rely on a backend to provide playback capabilities. LiSP currently have only a GStreamer backend.

A backend provide a “media object” to the cue, this object is composed by multiple elements that can be added/removed and configured, every element process data and feed the result to the next element. In GStreamer the set of active elements is referred as “pipeline”.

The elements and their settings can be changed in a specific tab in the cue settings:



The active elements can be changed using the *Change Pipeline* button



The default elements can be changed via `File > Preferences > GStreamer settings`

7.1 Input elements

Feed the pipeline with data

7.1.1 URI Input

Read and decode data from a URI (usually a URL), can be a local file or a remote one (eg: http://docs.gstreamer.com/media/sintel_cropped_multilingual.webm)

- **Source:** the URI to look for data (a “find file” button is provided for searching local files)
- **Buffering:** buffering options (for slower random access media such as a network file server)

- **Use Buffering:** enable buffering
- **Attempt download on network:** attempt to download the entire file on disk
- **Buffer size:** buffer size in bytes, -1 will use the default value

7.1.2 Auto Src

Get the audio from the system-default input device (eg: microphone), no option is provided

7.1.3 Preset Src

Generate some tone using some wired functions. Just Fun :-)

Don't try to use this in combination with the speed element or bad things will happen

Note: To use `Auto Src` and `Preset Src` you need to create a media cue with some random file, then change the source element.

7.2 Plugins elements

Used for audio-processing or data-probing, in some case the order affect the results

7.2.1 Volume

Allow to change the volume level, or mute the media.

- **Volume:** volume level in dB (can be muted)
- **Normalized Volume:** parameter used by other components (e.g. `ReplayGain`) to normalize the volume level without affecting user values, you can only reset the value (to 0dB).

7.2.2 10 Bands Equalizer

Allow to equalize the media with 10 frequency bands [30Hz-15KHz].

7.2.3 dB Meter

Allow external components to get the current sound level, used for UI visualization.

- **Time between levels:** millisecond between one extracted value and the next *lower values will use a more CPU*
- **Peak TTL:** Time To Live of decay peak before it falls back (in milliseconds)
- **Peak falloff:** Decay rate of decay peak after TTL (in dB/sec)

7.2.4 Speed

Speedup or slowdown the media, without affecting the pitch.

7.2.5 Pitch

Allow to change the media pitch by semitones.

7.2.6 Compressor/Expander

Provide Dynamic range compression.

- **Type**
 - *Compressor*
 - *Expander*
- **Curve shape: Selects how the ratio should be applied**
 - *Hard Knee*
 - *Soft Knee*
- **Ratio:** Ratio that should be applied
- **Threshold:** minimum value from which the filter is activated (in dB)

7.2.7 Audio Pan

Allow to control stereo panorama (left <-> right).

Note: When used the audio will be forced to stereo

7.2.8 Custom Element

Allow to manually create a custom GStreamer “elements” using the framework syntax, some instruction and example can be found [here](#).

7.3 Output elements

Push data to an output device

7.3.1 Auto sink

Use the system-default output device, no option is provided.

7.3.2 ALSA sink

Output to an ALSA device

- **ALSA device:** the output device to be used (parsed from asound configuration file)

7.3.3 PulseAudio sink

Output to the default pulseaudio output device, no option is provided.

7.3.4 Jack sink

Output to Jack server

An editable view of the current connections is shown, on the left the cue outputs, on the right the available jack inputs. Selecting one input and one output it's possible to connect/disconnect using the provided button.

Note:

- Unless the cue is created with `Jack sink` as output, by default all channels are disconnected
 - The connections to Jack are opened/closed when the cue start/stop
 - If no instance of a Jack-server is found a new one is started
-

Warning: This element can cause problems depending on the jack server configuration and/or status. Currently it's quite hard to debug those problems since the element is partially based on a GStreamer element that allow little control over the used client.

8 GStreamer Backend - Custom Elements

One of the most used functionality of GStreamer is the ability to create pipelines from a text description, usually this is done from a CLI interface (e.g. on a terminal) using the `gst-launch` program, in LiSP it's possible to create a custom media-element using this functionality.

8.1 Element Syntax

From this point `element (s)` refer to a GStreamer component and not to LiSP.

8.1.1 Properties

PROPERTY=VALUE

Sets the property to the specified value. You can use `gst-inspect` to find out about properties and allowed values of different elements.

8.1.2 Elements

ELEMENT-TYPE [PROPERTY_1 ...]

Creates an element of type *ELEMENT-TYPE* and sets its *PROPERTIES*.

8.1.3 Links

ELEMENT_1 ! ELEMENT_2 ! ELEMENT_3

The simplest link (exclamation mark) connects two adjacent elements. The elements are connect starting from the left.

8.1.4 Examples

The examples below assume that you have the correct plug-ins available. Keep in mind that different elements might accept different formats, so you might need to add converter elements like `audioconvert` and `audioresample` (for audio) in front of the element to make things work.

Add an echo effect to the audio:

```
audioecho delay=500000000 intensity=0.2 feedback=0.3
```

Add a reverb effect to the audio:

```
audioecho delay=200000000 intensity=0.4 feedback=0.45
```

Removes voice from sound (or at least try to do so):

```
audiokaraoke filter-band=200 filter-width=120
```

Remove voice from sound and (then) apply a reverb effect:

```
audiokaraoke filter-band=200 filter-width=120 ! audioecho delay=200000000  
intensity=0.4 feedback=0.45
```

Extracted from the [GStreamer SDK docs](#)

9 Command Cues Examples

Examples of Command cues to control other programs using LiSP:

9.1 LibreOffice

9.1.1 Impress

- **Start slideshow:** `xdotool key --window "$(xdotool search --class Libreoffice | head -n1)" F5` (requires xdotool)
- **Next slide:** `xdotool key --window "$(xdotool search --class Libreoffice | head -n1)" Right` (requires xdotool)

9.2 VLC

- **Playback a file (full-screen):** `vlc -f <file/path>`

9.3 MPV PLAYER

Allows to use a single mpv player instance, which is controlled through their json IPC interface. Use “Start Player” to init the player, you have to edit `--geometry` according to your display setup, “Load File” to play a Video file (you have to fill in the `$MEDIA` variable at the beginning of this command). The videos are played and stays open until next file is started. Images work too, so it is possible add add black images between the videos if needed.

- **Start Player:** `mpv --input-ipc-server=/tmp/mpvsocket --idle --keep-open=always --osc=no --fullscreen --geometry=1920:0`
- **Load File:** `MEDIA="<file/path>"; printf '{ "command": ["loadfile", "%s"] }\n' $MEDIA|socat - /tmp/mpvsocket`
- *A complete set of commands can be downloaded and imported as preset in the [GitHub releases page](#).*

10 ArtNet Timecode

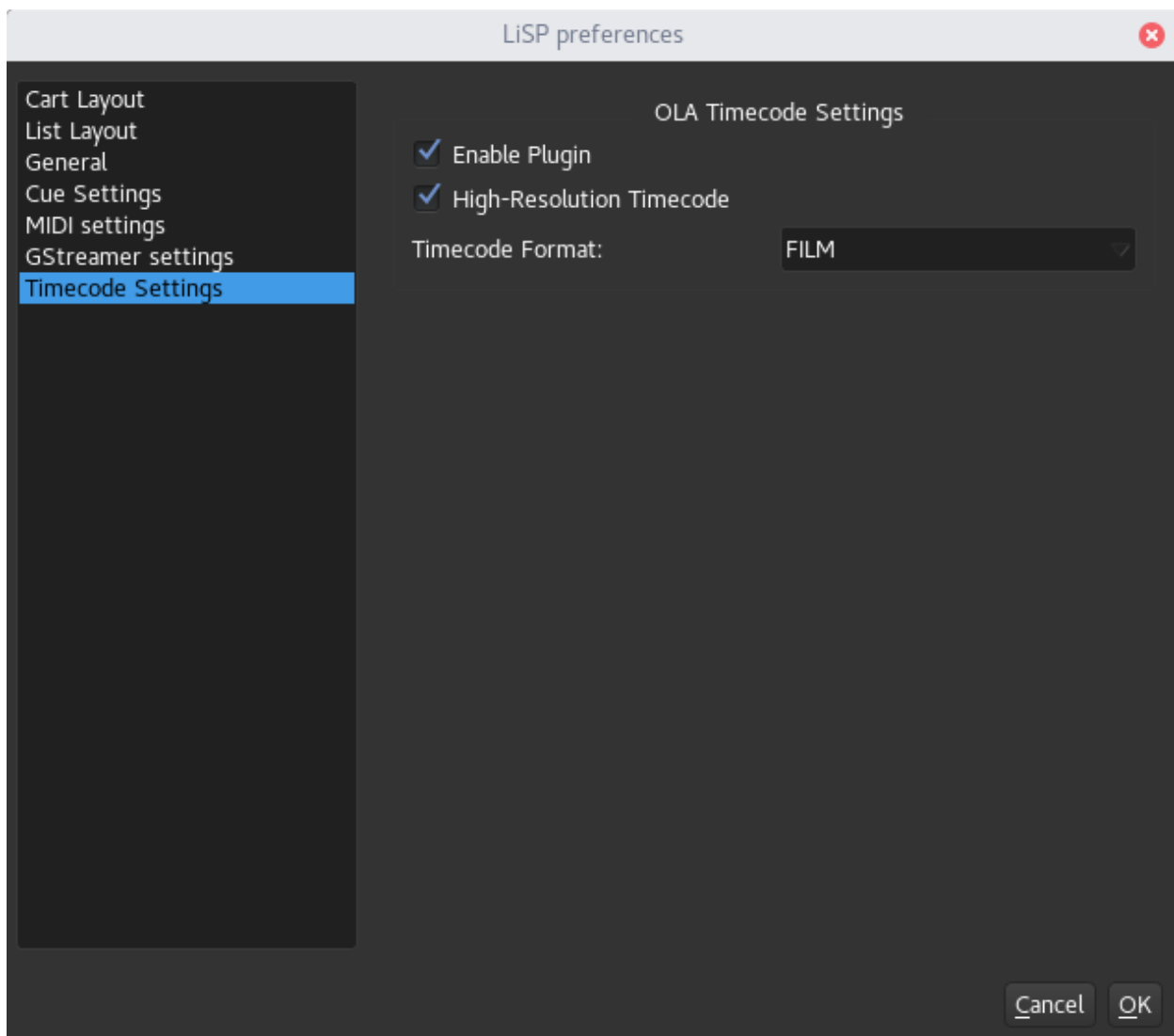
This plugin can be used to send the timecode of your running audio files over ArtNet to trigger cues inside a lighting control software or lighting desk which support ArtNet Timecode such as “Chamsys MagicQ”. This plugin works is meant as a alternative to the Chamsys Winamp plugin, which doesnt work under Linux. To get an general idea what is this all about, have a look [here](#).

In order to work, this plugin needs [OLA](#) installed and a running OLA session on your computer.

10.1 How to use

First make sure that OLA is up and running, <http://localhost:9090> gives you the OLA interface.

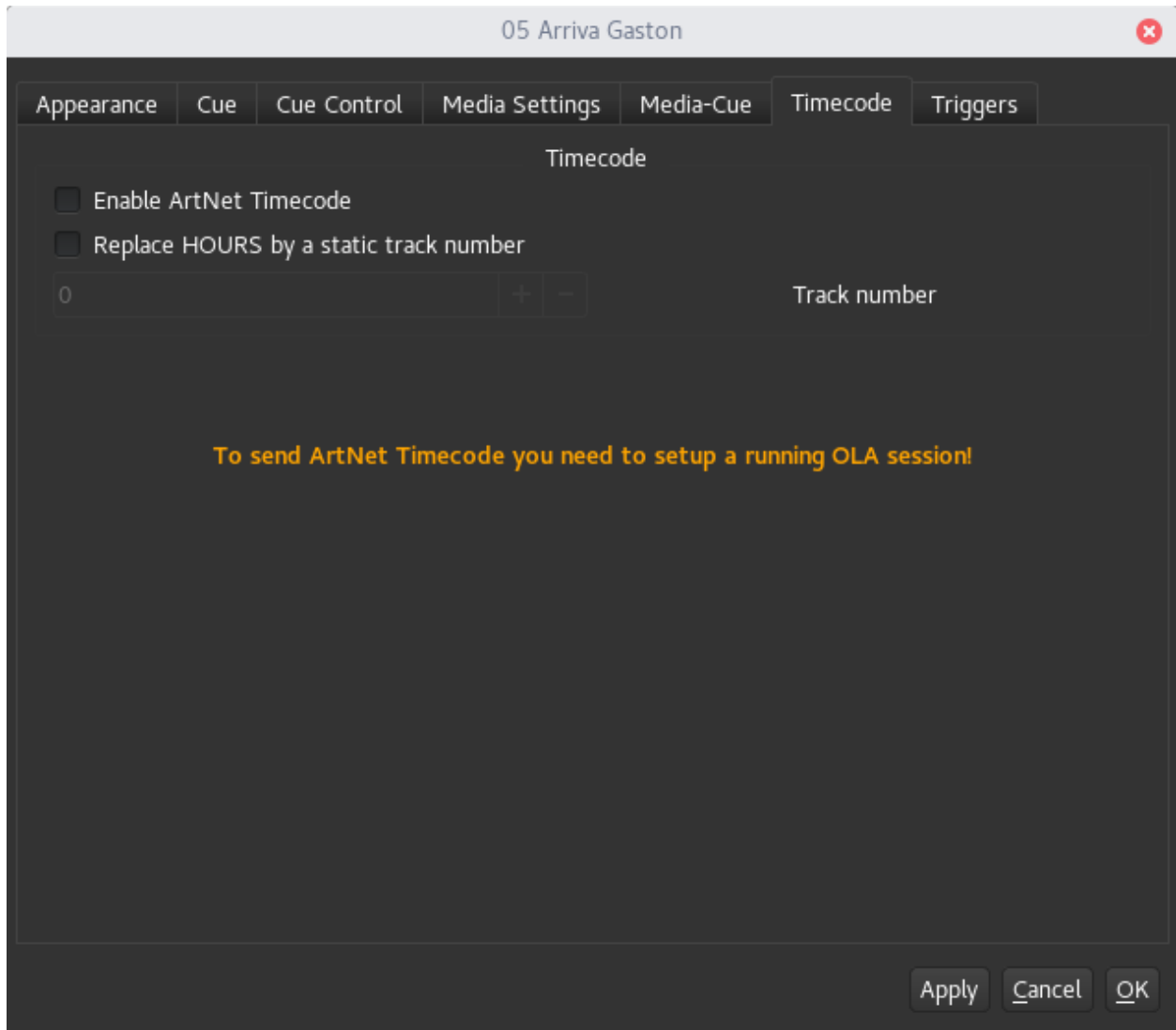
10.2 Timecode Preferences



To enable and setup the plugin go to `File > Preferences > Timecode Settings`.

- **Enable Plugin:** enables/disable the plugin
- **High Resolution Timecode:** enables sending every single frame, to get more accurate timing
- **Timecode Format:** choose between SMPTE, FILM and EBU. The format has to match the timecode used by the software which receives it.

10.3 Timecode Cue Settings



For media cues you can decide if it sends timecode or not. This can be set in the `Cue-Settings > Timecode`

Tab.

- **Enable ArtNet Timecode:** enables sending timecode for this cue
- **Replace HOURS by a static track number:** if checked, the `HOURS` field in the timecode is replaced by a static number, which can be used to identify which track currently sends timecode to your lighting software.

Note: If you work with multiple cuelists on your lighting desk you than can choose the following setup as example:

- cuelist 1 refers to track 1 and uses `HOUR=1`
 - cuelist 2 refers to track 2 and uses `HOUR=2`
 - ... and so on
-

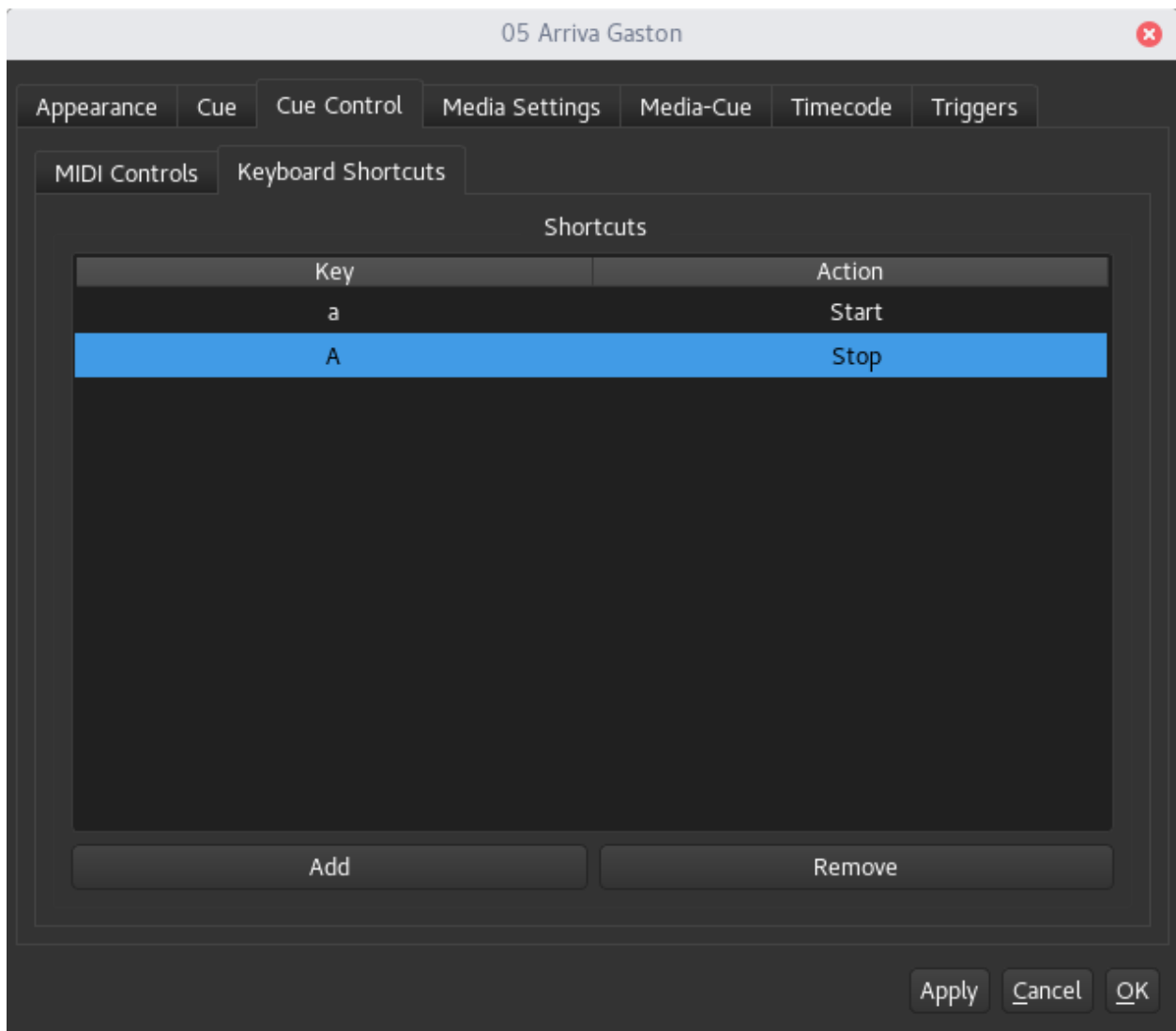
11 Cue Controls

Provide control over cues using keyboard and MIDI.

11.1 How to use

Settings are provided per cue, and are accessible through a specific tab in the cue settings window:

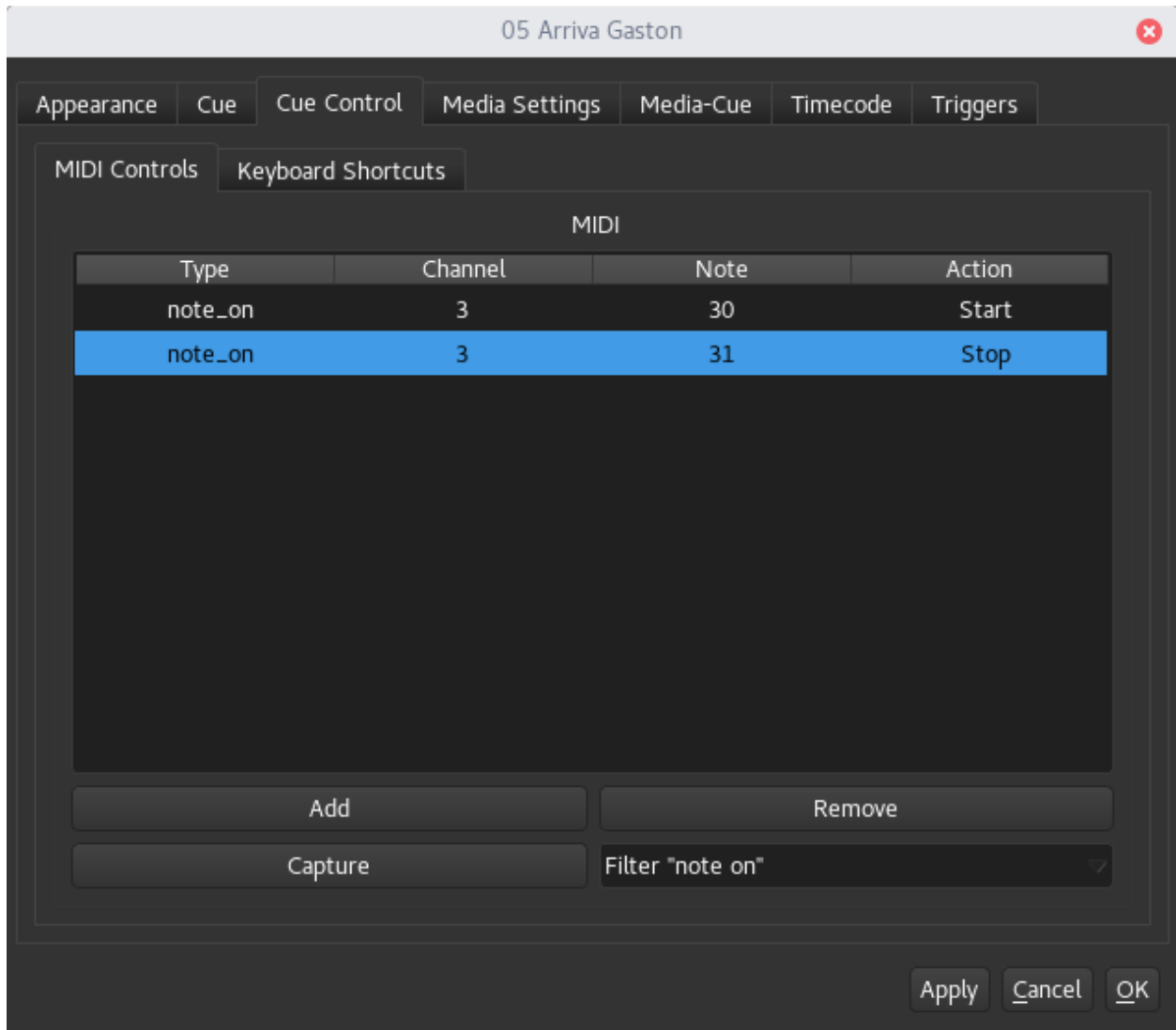
11.2 Keyboard



- **Key:** the key (character) that trigger the cue
- **Action:** the action to be executed when the key is pressed (Start/Stop/...)

New keys can be added/removed using the buttons at the table bottom. A key can be any single character that the keyboard is able to insert, so special keys are excluded, Upper/lower cases are considered, so “A” is not the same of “a”. In general, what is taken in account, it’s not the pressed key, but the typed character.

11.3 MIDI



- **Type:** The message type (only *note_on/off*)
- **Channel:** MIDI message “channel”
- **Note:** MIDI message “note”
- **Action:** Action to execute when a matching message is received
- *The MIDI Note “velocity” is ignored*

New MIDI messages can be added/removed manually using the provided buttons, or can be captured directly from the device, when doing so, a filter is provided to select the type of messages to be captured.

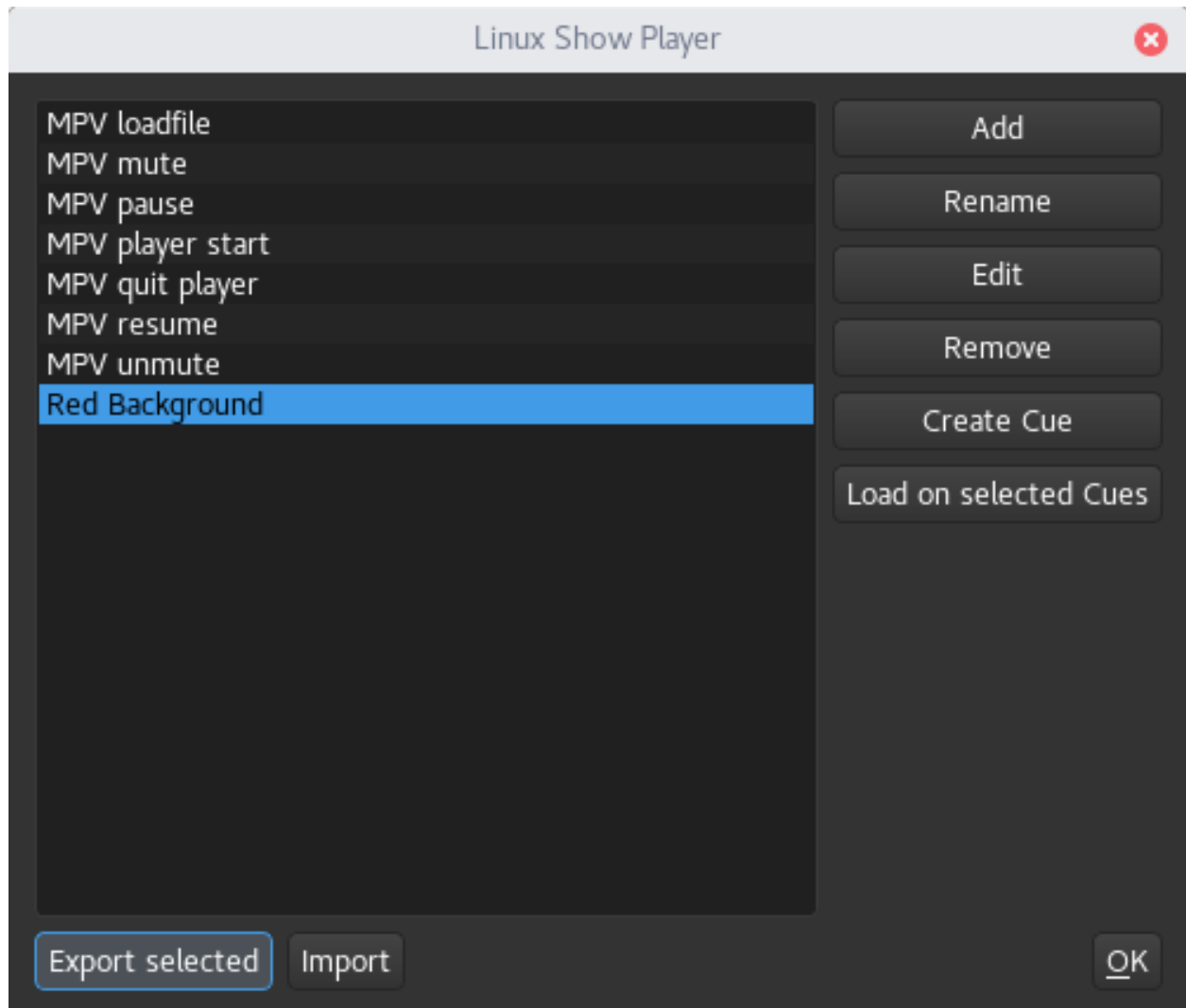
The used MIDI device can be changed in the application settings `File > Preferences > MIDI Settings`

12 Presets

Allow to create, edit, import and export presets for cues.

12.1 How to use

The main interface is accessible via `Tools > Presets`



On the left the list of the available presets (sorted by name), double-click to edit a preset. Multiple preset can be selected using the `CTRL` and `SHIFT`.

On the right a series of buttons gives access to the following:

- **Add:** allow to manually create a preset
- **Rename:** rename the selected preset

- **Edit:** edit the selected preset
- **Remove:** remove the selected preset
- **Create Cue:** create a cue from the selected presets
- **Load on selected Cues:** load a preset on selected cues

On the bottom:

- **Export selected:** export the selected presets to a custom archive
- **Import:** import from an exported preset

Note: The archive use a custom extension to easily filer others files, but it's a standard zip file.

The following options are provided in the cue context menu (right-click):

- **Load preset:** load a preset on the cue
- **Save as preset:** save the cue settings as a preset

Note: Preset are saved under `$HOME/.linux-show-player/presets/`

13 ReplayGain & Normalization

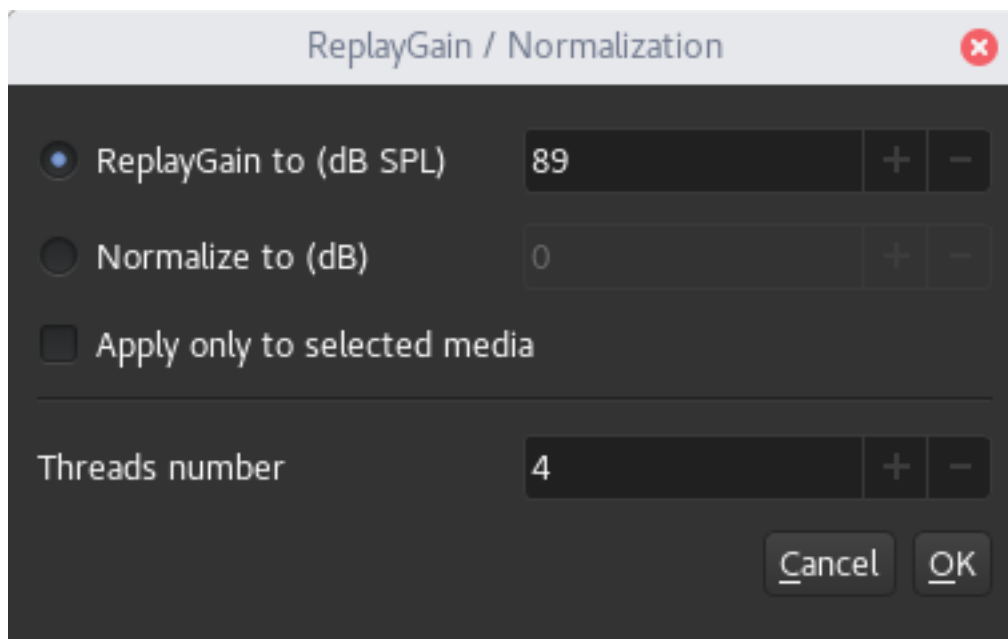
This module provide a simple utility to calculate a **Normalized / ReplayGain** volume for media cues. The values are used as *Normalized Volume* for the `Volume` media-element, if a **ReplayGain** value is already stored in the media-file metadata (e.g ID3 tags) it will be used.

Note: The original files are left untouched.

13.1 How to use

Via `Tools > ReplayGain / Normalization` menu the following options are provided:

- **Calculate:** Open a dialog to set some option and start the calculation
- **Reset all:** Reset to 0dB the normalized volumes of all cues
- **Reset selected:** Reset to 0dB the normalized volumes, only for the selected cues



- **ReplayGain:** Use ReplayGain normalization using the reference value in dB SPL (89 is the standard default)
- **Normalize:** Use a simple normalization to the reference value in dB (0 is the maximum value)
- **Only selected cues:** apply only to the currently selected cues
- **Thread number:** Number of concurrent/parallel calculations (default to the cup cores)

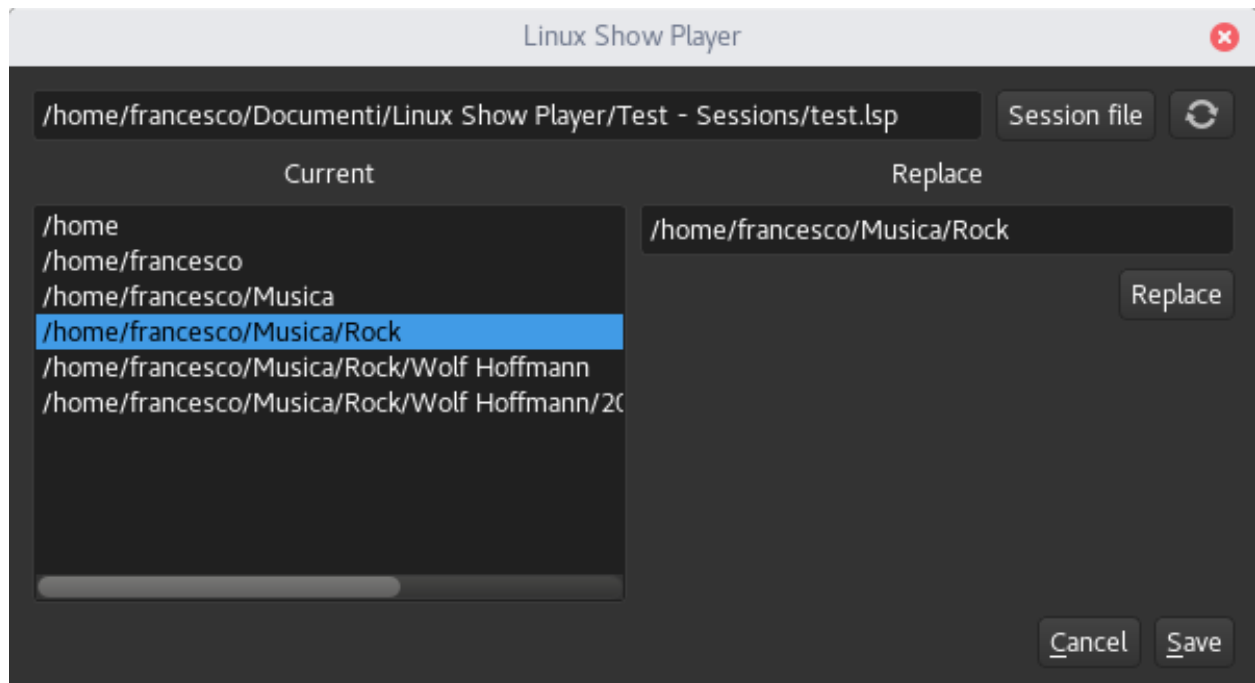
Note: that the process may require some time, depending on the CPU, the number and size of the files involved.

14 Session URI Editor

This module provide a simple utility that allow to replace the media-files “paths” stored into a session file, this allow to move a show to a different PC with media-files located in different directories.

14.1 How to use

In the menu `Tools > Uri session change` open the following window:



On the top, the `Session file` button allow to search for a file, on the right a `reload` button allow to reset all the changes.

On the left, a list of (split) directories is shown, those can be selected, and edited, when the `replace` button is pressed the list is updated to reflect the changes.

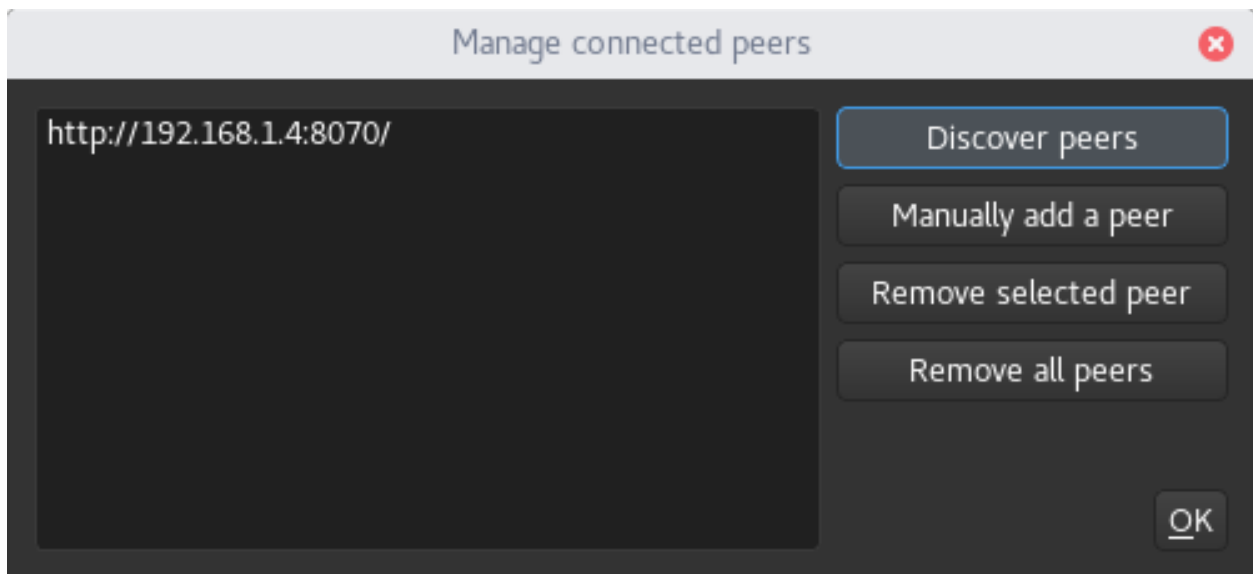
The `save` button will allow to save the session file to a new location.

15 Synchronization

The goal of this plugin is to allow two or more LiSP sessions running on different PCs to be synchronized during a performance (no edit synchronization). The two sessions are supposed to be identical, so, only user interaction with the cues are replicated on other sessions.

15.1 How to use

The plugin usage is quite simple, the only thing to do, is to add the remote session you want to “control” to the peer-list, this can be done via `Tools > Synchronization > Manage connected peers`



On the left you can find the peers list, on the right the following buttons:

- **Discover peers:** Allow to search for other sessions in the network;
- **Manually add peer:** Allow to manually add a peer, using its IP address;
- **Remove selected peer:** Remove the selected peer;
- **Remove all peers:** Remove all the peers.

To easily retrieve the (local) IP address `Tools > Synchronization > Show your IP` will display the current IP address of the PC.

15.2 How it works

Once a session is connected, user actions are replicated on the connected one. This is achieved by sending some info over the network, by default the 8070 and 50000 (for the discovery) ports are used, those values can be changed in the configuration file `$HOME/.linux-show-player/config.cfg`.

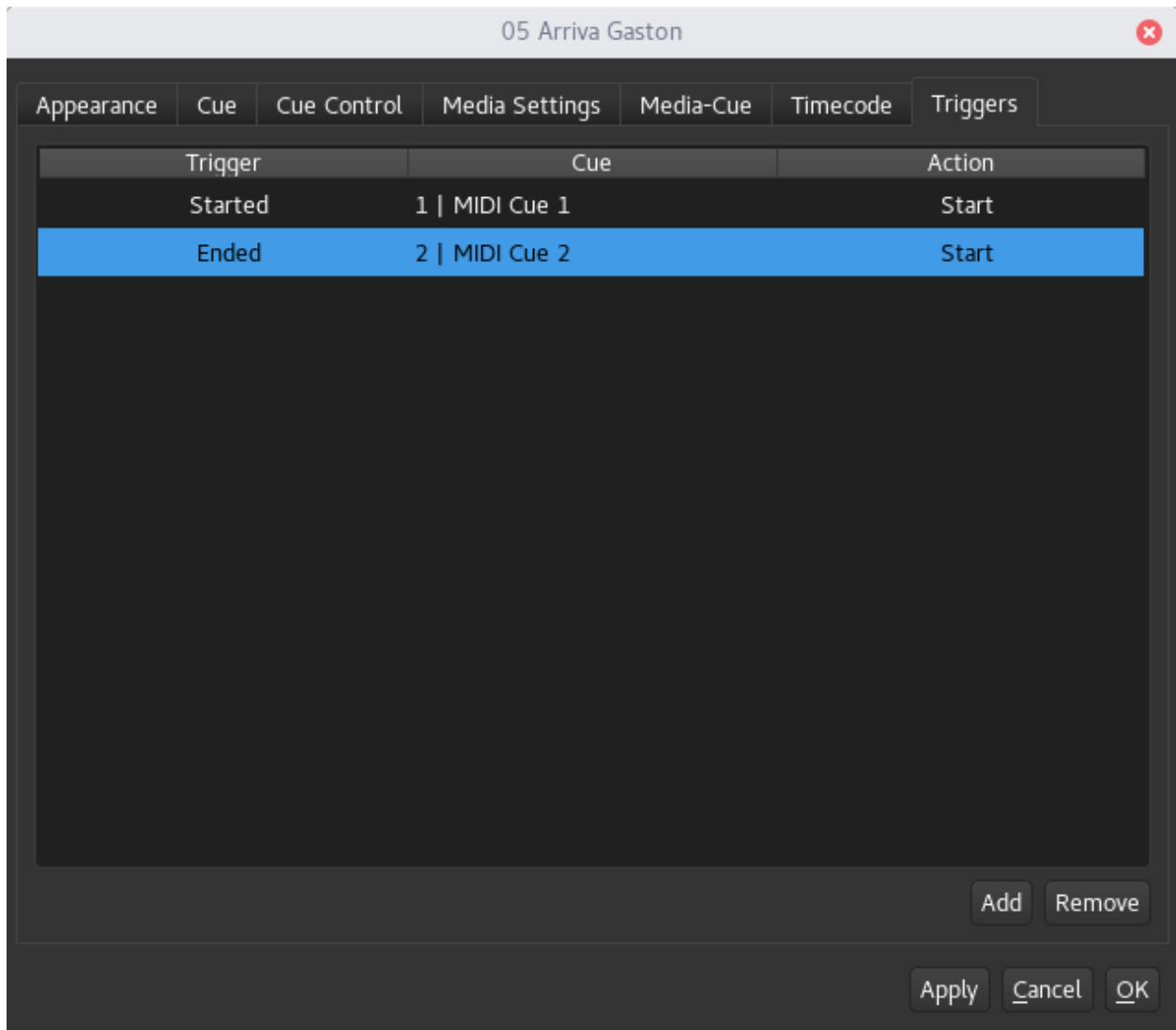
Two or more sessions can be mutually connected, this way all the sessions share the same “state”.

16 Triggers

This plugin allow to create triggers based on cues events, and to assign them a specific action on another cue.

16.1 How to use

Triggers can be added/removed and modified from the cue-settings dialog of the cue from which the event is generated (source):



16.2 Events

The following events are available:

- **Started:** The cue is started (after pre-wait)

- **Stopped:** The cue is stopped
- **Paused:** The cue is paused
- **Ended:** The cue is ended