Q Light Controller Plus

User Documentation

Updated to version 4.8.5
April, 12th 2015

- **Basics**
  - Basic Concepts & Glossary
  - Questions & Answers
- **Advanced**
  - Command-Line Parameters
  - Manual parameters tuning
  - GUI style customization
  - Kiosk mode
  - Web Interface
- **Main Window**
  - DMX Monitor
  - DMX Address tool
  - DMX Dump
  - Live Edit
- **Fixture Manager**
  - Add/Edit Fixtures
  - Fixture Group Editor
  - Channels Groups Editor
  - Fixtures remapping
  - How to add fixtures
- **Function Manager**
  - Scene Editor
  - Chaser Editor
  - Show Editor
  - EFX Editor
  - Collection Editor
  - RGB Matrix Editor
    - RGB Script API
  - Script Editor
  - Function Wizard
  - Select Functions

- Select Fixtures

- ## Show Manager

- ## Virtual Console

  - Button
  - Button Matrix
  - Add Button Matrix
  - Slider
  - Slider Matrix
  - Animation
  - Speed Dial
  - XY Pad
  - Cue List
  - Frame
  - Solo Frame
  - Label
  - Audio triggers
  - Select Input Channel
  - Widget Styling & Placement

- ## Simple Desk

- ## Input/Output

  - Input/Output mapping
  - Input profiles
  - Audio Input/Output

  Plugins:

  - ArtNet
  - DMX USB
  - E1.31 (sACN)
  - Enttec Wing
  - HID
  - MIDI
  - OLA
  - OSC
  - Peperoni
  - uDMX
  - Velleman
  - Loopback

- ## Fixture Definition Editor

  - Capability Editor
  - Capability Wizard
  - Channel Editor

- Fixture Editor
- Mode Editor
- Head Editor

- **Tutorials**

  - Scene tutorial
  - Multipage frame tutorial
  - Sound control tutorial
  - BCF2000 + LC2412 tutorial

Basic Concepts & Glossary

Q Light Controller Plus (QLC+ for short) is meant to control lighting equipment used in various performances, like live concerts and theatres etc. The main intention is that QLC+ will be able to outperform commercial lighting desks without the need for a 500+ page manual through the use of an intuitive and flexible user interface.

This page has been arranged in alphabetical order to facilitate searching for a specific topic.

Audio

An audio function is an object representing an audio file stored in a disk.
QLC+ supports the most common audio formats like Wave, MP3, M4A, Ogg and Flac. It supports mono or stereo channels and several sample rates like 44.1KHz, 48KHz, etc...
Audio functions can be placed in Chaser or in a Show at the desired time, using the Show Manager panel.
Like most of the QLC+ functions, Audio supports fade in and fade out times.

Warning: Even though QLC+ allows you to, it is not possible to play two audio files simultaneously. Especially on Windows, you might experience unwanted crashes.

Blackout

Blackout is used to set all channels in all universes to zero and keep them that way, regardless of which functions are currently being run or what values have been set to them manually. When blackout is turned off, the current values of all channels are sent to each universe.

Capabilities

Some channels in intelligent fixtures provide many kinds of functions, or *capabilities* like switching the lamp on when the channel value is [240-255], setting a red color on a color wheel when the value is exactly [15], or simply controlling the fixture's dimmer intensity with values [0-255]. Each of these individual function is called a capability and each of them has these three properties:

- Minimum value: The minimum channel value that provides a capability.
- Maximum value: The maximum channel value that provides a capability.
- Name: The friendly name of a capability.

Channel Groups

Channel groups can be added and defined in the Fixture Manager panel by using the Channel groups editor.
Channel groups can have a user defined name and can group together any user defined channels selected from an existing Fixtures list.

Chaser

A chaser function is built up from multiple scenes that are run in sequence, one after the other, when the chaser function is started. The next function is run only after the previous has finished. Any number of functions can be inserted to a chaser.

The Chaser function's direction can be reversed. The Chaser function can also be set to do an infinite loop, an infinite ping-pong-loop (direction is reversed after each pass) or it can run through just once, in a single-shot mode, after which it terminates by itself. If the function is set to loop infinitely, it must be stopped manually.

As of version 3.3.0, each chaser has its own speed settings:

- **Fade In:** The fade in speed of a step
- **Hold:** The hold time of a step
- **Fade Out:** The fade out speed of a step
- **Duration:** The duration of a step

Copies of chaser functions can be created with the Function Manager. The scenes inside a chaser are not duplicated when a chaser is copied. Only the order and direction are copied to the new one.

Click And Go

Click And Go is a technology that allows the user to quickly access macros and colors in a completely visual way and with just a couple of clicks. This can lead to more efficient live shows and more freedom to choose the desired result very easily.
So far, three types of widgets are available:

- Single Color (applies to: Red, Green, Blue, Cyan, Yellow, Magenta, Amber and White intensity channels)
- RGB Color Picker. Controls values for selected RGB channels with a single click
- Gobo/Macro Picker. Access and display a Gobo/Macro defined in the Fixture definition

An overview with screenshots is available here

Collection

A collection function encapsulates multiple functions that are run simultaneously when the collection function is executed. Any number of functions can be inserted to a collection, but each function can be inserted only once and a collection cannot be a direct member of itself.

Collections have no speed settings. The speed of each member function is set individually using their own editors.

Copies of collection functions can be created with the Function Manager. The functions contained in a collection are not duplicated; only the list of functions is copied.

DMX

DMX is short for **D**igital **M**ultiple**X**. It basically defines a whole bunch of properties, protocol, wiring etc. In the case of lighting software, it defines the maximum number of channels (512) per universe and the value range of each channel (0-255).

QLC+ supports 4 individual universes. They do not necessarily need to be connected to DMX hardware; rather, DMX has just been selected as the de facto lighting standard. Actual hardware abstraction (whether it's analogue 0-10V, DMX or some other method) is achieved through output plugins.

EFX

An EFX function is mainly used to automate moving lights (e.g. scanners & moving heads). The EFX can create complex mathematical paths on an X-Y plane that are converted to DMX values for the fixture's pan and tilt channels. **Only fixtures that contain valid pan & tilt channels can take part in an EFX function.**

Fixtures

A fixture is essentially one lighting device. It can be, for example, one moving head, one scanner, one laser etc.. However, for simplicity, individual PAR cans (and the like) that are usually controlled thru one dimmer channel per can, can be grouped together to form one single fixture.

With the Fixture Definition Editor, users can edit shared fixture information stored in a fixture library that contains the following properties for each fixture:

- Manufacturer (e.g. Martin)
- Model (e.g. MAC250)
- Type (Color Changer, Scanner, Moving Head, Smoke, Haze, Fan...)
- Physical properties (bulb type, beam angle, dimensions...)
- Channels:
    - Channel group (Intensity, Pan, Tilt, Gobo, Color, Speed etc.)
    - 8bit and 16bit channel bindings for pan & tilt groups
    - Optional primary color for intensity channels (RGB/CMY)
    - Value ranges for channel features (e.g. 0-5:Lamp on, 6-15:Strobe etc..)

These fixture definitions can then be used to create actual fixtures in the Q Light Controller Plus application, that will have additional properties defined by users:

- DMX Universe
- DMX Address
- Name

Several instances of a fixture can be created (e.g. users must be able to have several instances of a MAC250 in use). Each fixture can be named, but the name is not used internally by QLC+ to identify individual fixture instances. The same goes for the DMX address. Nevertheless users are encouraged to name their fixtures in some systematic way to help identify each of them -- if necessary.

Generic dimmer devices don't need their own fixture definitions, because usually multiple dimmers are patched into a common address space, employing one or more dimmer racks. Users can create instances of these generic dimmer entities just by defining the number of channels each one of them should have.

Fixture Group

A fixture group is, as the name says, a group of fixtures. They also define (at a rather basic level) the actual physical, real world arrangement of these fixtures. This knowledge can be used, for example, in the RGB Matrix to produce a wall of RGB-mixable lights that can act as individual pixels in a graphic pattern or scrolling text.

Fixture Mode

Many manufacturers design their intelligent lights in such a way that they can be configured to understand different sets of channels. For example, a scanner might have two configuration options: one for only 8bit movement channels (1x pan, 1x tilt) and another one for 16bit movement channels (2x pan, 2x tilt). Instead of creating a completely new fixture definition for each variation, they have been bundled together in QLC+'s fixture definitions into fixture modes.

Functions

The number of functions is practically unlimited. Functions are used to automate the setting of values to DMX channels. Each function type has its own way of automating lights.

The function types are:

- Scene
- Chaser
- Sequence
- EFX
- RGB Matrix
- Collection
- Show
- Audio

Each function can be named and, although the name is not used to uniquely identify individual functions, users are encouraged to name their functions in some systematic and concise way to help identify each of them. For your own comfort.

As of version 3.3.0, each function has its own speed settings:

- **Fade In:** The time used to fade HTP (in Scenes also LTP) channels to their target value
- **Fade Out:** The time used to fade HTP/intensity channels back to zero
- **Duration:** The duration of the current step (not applicable on Scenes)

Grand Master

The Grand Master is used as the final master slider before values are written to the actual physical DMX hardware. Usually, the Grand Master affects only **Intensity** channels, but can also be changed to effect the values of **all** channels.

The Grand Master has also two **Value Modes** that control the way *how* the Grand Master affects channel values:

- Reduce: Affected channels' values are reduced by a percentage set with the Grand Master slider. For example, Grand Master at 50% will result in all affected channels being reduced

to 50% of their **current** values.

- Limit: Affected channels cannot get larger values than the value set with the Grand Master slider. For example, Grand Master at 127 will result in all the affected channels' maximum values being limited at exactly 127.

Head

A head represents an individual light output device in a fixture. Usually, a single fixture contains exactly one output, like the lens, the bulb, or a set of LEDs. There is, however, an increasing number of fixtures on the market that, although treated as a single fixture, have multiple light output devices, i.e. heads.

For example, you might have a RGB LED bar fixture that is assembled onto a single chassis and as such it appears as a single fixture with one DMX input and one DMX output. However, it is actually comprised of four separate RGB LED "fixtures". These separate fixtures are treated in QLC+ as heads; they share some properties with their sibling heads, they can be controlled individually, but they might also have a master intensity control that controls the light output of all the heads together.

Each head belongs to a Fixture Mode because in one mode, a fixture might provide enough channels to control each of its heads individually while in another mode, only a handful of channels might be provided for controlling all the heads simultaneously.

HTP (Highest Takes Precedence)

HTP is a rule that decides what level is sent to out to a DMX universe by a channel when the channel is being controlled by more than one function or Virtual Console widget. Generally, intensity channels obey the HTP rule. This includes generic intensity channels used to control *light intensity* with dimmers and also channels controlling the intensity of a color, typically in an LED fixture.

The HTP rule is simple: the highest level (nearer 100%) that is currently being sent to a channel is the one that gets sent out to the DMX universe.

Let's say you have two sliders that control the same intensity channel. First, you set slider 1 to 50% and then move slider 2 from 0% to 75%. As long as slider 2 is below 50% nothing happens, but after crossing the 50% level set by slider 1, the light intensity increases up to 75%. If you drag slider 2 again towards 0%, the light intensity decreases until it reaches the 50% set by slider 1 and stays at 50% until slider 1 is dragged down.

A crossfade between 2 Scenes will replace the HTP levels in the first scene with the HTP levels of the second. The new HTP levels will be combined with HTP levels from other functions and virtual console widgets as above. See also LTP.

Input/Output plugins

QLC+ supports a variety of plugins to send and receive data from/to the external world.
A plugin can be an interface to physical devices (such as DMX adapters or MIDI controllers) or to a network protocol (such as ArtNet, OSC or E1.31).
Plugins support input, output or feedback capabilities depending on the device or the protocol they're controlling.

The primary input methods for QLC+ are naturally the keyboard and mouse. Users can assign keyboard keys to virtual console buttons and drag sliders and do pretty much everything with a mouse.

Although, with plugins it is possible to attach additional input devices to one's computer to alleviate the rather clumsy and slow user experience that is achieved with a regular mouse and a keyboard. Plugins supporting an input line provide capabilities for getting external devices to produce input data to various QLC+ elements.

An input line is a connection provided by some hardware or network which is accessed through an input plugin. It can be, for example, a MIDI IN connector in the user's computer (or peripheral) to which users can connect MIDI-capable input devices like slider boards etc.

An output line is a connection provided by a hardware or network which is accessed through an output plugin. In other words, it is a real DMX universe, but has been dubbed output to separate it from QLC+'s internal universes. You could think of them as individual XLR output connectors in your DMX hardware.

Input profiles

Input profiles can be thought of as fixtures' cousins; they contain information on specific devices that produce input data. An input device can be, for example, a slider board like the Behringer BCF-2000, Korg nanoKONTROL, an Enttec Playback Wing...

LTP (Latest Takes Precedence)

LTP is a rule that decides what level is sent to out to a DMX universe by a channel when the channel is being controlled by more than one function or Virtual Console widget. Generally, it is used for channels that have been assigned to groups other than the **Intensity** group, such as pan, tilt, gobo, strobe speed and other *intelligent fixture parameters*

The LTP rule is simple: the latest level that has been set by a function or a Virtual Console widget gets sent out to the DMX universe.

During a crossfade between Scenes, LTP levels will often be changed. This has to be handled with some care as some LTP levels need to jump immediately to a new level, for example, changing from one gobo to another. LTP groups such as pan and tilt, however, might need to change gradually from one level to another during a crossfade. Different timings can be achieved by combining scenes in a Collection. See also HTP.

Modes

Q Light Controller Plus is based on the common concept of having two distinct operational modes to prevent accidental and possibly harmful changes during operation:

- Design mode is meant to edit the behaviour of the program; create and edit functions and fixtures and adjust how they work.
- Operate mode is meant to execute the created functions that eventually control the user's lighting fixtures.

RGB Matrix

An RGB matrix function can be used to impose simple graphics and text on a matrix (a grid or a wall) of RGB and/or monochrome fixture heads. The RGB matrix function has been designed to be extendable with scripts that can be written by users.

Each RGB matrix has its own speed settings:

- **Fade In:** Time to fade each pixel ON
- **Fade Out:** Time to fade each pixel OFF
- **Duration:** The duration of the current step/frame

RGB Script

An RGB script (not to be confused with the) is a program written in ECMAScript (also known as JavaScript) that produces the necessary image data for RGB Matrix functions. Learn more from the RGB Script API page.

Scene

A scene function comprises the values of selected channels that are contained in one or more fixture instances. When a scene is started, the time it takes for its channels to reach their target values depends on the scene's speed settings:

Each function has its own speed settings:

- **Fade In:** The time used to fade all channels to their target values, from whatever value they had
- **Fade Out:** The time used to fade HTP/intensity channels back to zero. Note that ONLY HTP channels are affected by this setting.

Copies of scene functions can be created with the Function Manager. All of the scene's contents are copied to the duplicate.

Sequence

A Sequence has some of the functionality of a Chaser.
It is equivalent to a Chaser in which each step is a single Scene and every one of those Scenes controls the same set of channels. A Sequence is bound to one specific Scene, which means that all the steps of the Sequence can only control the enabled channels of that Scene.
When creating new steps in a Sequence, no Function selection pop-up will appear, since a Sequence step cannot include other Functions, unlike a Chaser step.
When a Sequence is created, a special sequence icon will appear in the Function Manager as a child of the Scene to which it is bound.
To understand the difference between a Sequence and a Chaser, you are invited to read the second paragraph of the Show Manager documentation.

Script

The Script function works on a simple yet powerful scripting language to automate QLC+ functionalities in a sequential order. A Script can be modified with the Script Editor.

Show

A Show is an advanced function which encapsulates most of the QLC+ Functions to create a time driven light show. A Show can be created only with the Show Manager and can be inspected and renamed with the Show Editor.

Questions & Answers

In this page you will find the common questions that may come to mind when starting with QLC+. Here you can either find the answer directly or find help to point you in the right direction.

Q:	QLC+ cannot detect my USB device
A:	QLC+ supports a wide variety of USB devices. First of all you should check if the physical connection is OK. Usually a LED on your device should indicate if it is powered up and working correctly. If you are using Windows and your device is manufactured by Peperoni or Velleman, please read the information on how to get them working on these help pages. For licensing issues they both need an extra DLL file to work. Please check Peperoni output plugin or Velleman output plugin If you're using Linux, please check if your distribution detected the device when plugged in. Basically, the "`dmesg`" command should tell you something.

Q:	I've got several buttons in my Virtual Console. I need a way to disable the currently active button when I enable another one. How do I do that ?
A:	Simply place your buttons inside a Solo Frame. It does exactly that.

Q:	When I start an EFX function, all my fixtures go to full intensity and I can't make them dimmer.
A:	Please, read this note.

Q:	I just upgraded my Mac to OSX Mavericks but my Open USB DMX adapter doesn't work anymore.

A:	The problem is in a new Apple USB driver. The procedure to disable it follows below. **Please note that this can compromise the behaviour of other USB devices, so do it only if you know what you're doing!**
	From a terminal type the following commands:
	`cd /System/Library/Extensions/IOUSBFamily.kext/Contents/PlugIns`
	`sudo mv AppleUSBFTDI.kext AppleUSBFTDI.disabled`
	`sudo touch /System/Library/Extensions`

Q:	Where is the QLC+ user folder located in my system ?
A:	The user folder is where user fixtures, input profiles, RGB scripts and MIDI templates go. It changes depending on your operating system: **Linux**: it is a hidden folder in your user home directory: $HOME/.qlcplus **Windows**: it is a folder in your user (e.g. MyUser) directory: C:\Users\MyUser\QLC+ **OSX**: it is located in your user "Library" directory: $HOME/Library/Application\ Support/QLC+ You can reach any of these folders from a terminal with the 'cd' command. For example: `cd $HOME/Library/Application\ Support/QLC+` Please keep in mind that fixures and input profiles found in the user folder will have precedence over the same files in the QLC+ system folder.

Main Window

The QLC+ Main Window consists of three main parts:

1. A Menu bar containing buttons for global functions
2. Active panels selected by their respective tabs
3. Tabs that allow the selection of one of the QLC+ panels

Most of the controls throughout the software have tooltips, which pop up after holding your mouse cursor over them for a short while.

Menu bar (1)

The menu bar on the top of the workspace window contains the following buttons (from left to right):

New workspace

Open an existing workspace (hold the button longer for a list of recent files)

Save the current workspace

Save the current workspace with a new name

DMX Monitor

DMX Address tool

Audio triggers

Toggle full screen mode

View this documentation

View information about QLC+

DMX Dump

Live Edit: allows you to modify a function while QLC+ is in Operate Mode

Live Edit Virtual Console: allows you to modify the Virtual Console while QLC+ is in Operate Mode. Click to Toggle

Stop All Functions

Toggle Blackout

Switch between Design Mode and Operate Mode

Active Panel (2)

Everything happens here. Which panel is active depends on which tab is currently selected.

Panel Tabs (3)

At the bottom of main window you can find easy eye catching icons for switching between QLC+ panels that appear as follows (from left to right):

View the Fixture Manager

View the Function Manager

View the Show Manager

View the Virtual Console

View the Simple Desk

View the Input/Output configuration manager

DMX Monitor

The DMX Monitor is a useful tool to track the values that are being sent to the output universes. Only the information related to the required fixtures are displayed. The monitor's display options have no effect on actual fixture addressing, after all, it is just a **monitor**.
The DMX monitor has two display modes: **DMX view** and **2D view**.

DMX view

The DMX view shows the all the fixtures of the project, representing each channel with numbers and icons. It basically represents each channel in 3 rows:

- The channel group icon
- The channel number
- The channel value

Toolbar controls

2D View	By clicking on this button, it is possible to switch to the 2D view mode.
	Change the monitor font. To prevent the numbers from flickering and jumping, you should choose a proportional (i.e. fixed width) font; for example Monaco, Andale or Courier. The font property is global, meaning it will not be saved into your current project, but it will be stored in the QLC+ main configuration.
DMX Channels	Display fixtures' channel numbers as absolute DMX channels; channel numbers go from 1 to 512 as they are assigned to each fixture.
Relative Channels	Display fixtures' channel numbers relative to fixtures i.e. every fixture's channel numbers always start from 1.
DMX Values	Display channel values as absolute DMX values (0-255).
Percent Values	Display channel values as percentages of 255 (0-100%).
Universe	Select which universe to monitor. The first entry is always "All universes"

2D view

The monitor 2D view is another way to represent the fixtures of your project that, instead of numbers, uses a graphical preview trying to represent as much as possible the real result of a light-emitting device.

Currently, monitor reflects:

- Master Dimmer

- R/G/B
- C/M/Y
- Color wheels, if they contain RGB color value. Two-color values are not supported
- Shutter, which is open, unless the capability name contains "close" or "blackout" (example: "Shutter Close")

In 2D view mode it is possible to select which fixtures to display and their position in a grid representing the dimensions of a real stage.
The grid would like to reproduce the front view of a stage, but you can use it as a generic space as you might like.
Graphical items can be manually moved by dragging them over the grid, or, when clicked, they can be configured with the Monitor Fixture Editor panel that will be displayed on the right side of the window.

Toolbar controls

DMX View	By clicking on this button, it is possible to switch to the DMX view mode.
Grid dimensions	Set the width and the height of the 2D view grid by changing the values displayed in the two spin boxes
Grid units	Set the 2D view grid measurement units by selecting the desired one from the drop down menu. Possible options are meters and feet.
	Add a fixture to the 2D view grid. When clicking on this icon, the fixture selection dialog will be displayed. Fixtures already added to the view will be greyed as it isn't possible to add the same fixture twice
	Remove a fixture from the 2D view grid. Clicking on this icon will remove the currently selected fixture. A fixture is highlighted in yellow when selected.
	Open the Monitor background picture selection dialog. Here it is possible to choose between 3 possible modes: • **No background**: the Monitor 2D view won't have any background picture • **Common background**: the Monitor 2D view will display the chosen background picture • **Custom background list**: in this mode it is possible to select a background picture for a specific QLC+ Function. Just click on the and buttons to add/remove Functions and their associated background picture to the Monitor. When a Function in this list will start the Monitor 2D view background picture will change accordingly.
	Show/hide fixtures' names underneath their graphical representation

Fixture Item Editor

When a fixture is clicked, it gets highlighted in yellow and the Monitor Fixture Item Editor is

displayed on the right side of the window.
Following, the possible parameters that it is possible to tune:

Horizontal position	Set the position on the grid X axis using the grid measurement units
Vertical position	Set the position on the grid Y axis using the grid measurement units
Rotation	Set the rotation angle of the graphical item representing the selected fixture
Color gel	Set the color gel to be applied to the currently selected fixture item. This is useful for generic dimmers with a traditional light bulb, not emitting any color by itself. This function has no effect on RGB LED fixtures.
	Reset a previously set color gel

Pan/Tilt display

PAN/TILT display is schematically displayed using colored rings/arcs around fixture. Purple arc shows **P**AN angle, and Turquoise arc shows **T**ILT angle. Gray dots show PAN/TILT ranges. Zero (middle of the range) is at the bottom.

In the folowing picture PAN range is 660 degrees and tilt range is 300 degrees. Pan is at counter-clockwise end (-330 deg) and tilt is at -150 deg.

DMX Address Tool

The DMX address tool is a quite easy/self explanatory functionality of QLC+ introduced in version 4.3.2.
It helps you to calculate your fixtures addresses quickly in a visual way.
A representation of a common 10way DIP switch is displayed and you can select the background color, the horizontal and vertical orientation, so that the DIP switch is displayed exactly as you would see it on your fixture.

Note particularly the following, which applies to the 10th switch. Usually, it is used to enable/disable the DMX feature of a fixture, so if your product manual says it has to be raised to enable the DMX control, then remember to do it even if QLC+ displays it set to OFF.

DMX Dump

The DMX Dump functionality allows you to save the current DMX values that are being sent to the output universes at a particular moment. Basically it takes a "snapshot" of DMX channels and saves them for a later use.
DMX Dump can save values to a new Scene and also add the newly created Scene into an existing Chaser, Virtual Console button or slider

Please note that:

- Selected Channels/Chasers will be remembered when re-opening this window
- The DMX values saved are taken before the Grand Master

When opening the DMX Dump window, the following options will appear:

Scene name	Defines the name of the Scene that will be created. If no name is specified, a default name like "New Scene From Live" and a numeric identification will be set, allowing fast use of this functionality.
Dump all channels	If this option is selected, QLC+ will dump all the channels of all the universes and all the fixtures. To inform the user about what this option will do, a report in the form of (Universes, Fixtures, Channels) will be displayed.
Dump selected channels	If this option is selected, the panel below will be activated, allowing you to choose exactly which channels that you want to be saved into a Scene. They are organized in a tree by Universes and Fixtures.
Add to	This section offers you immediate use of your scene in an existing Virtual Console widget. Options are: • **Chaser**: The list contains all the chasers present when DMX Dump window is opened. Each Chaser has a checkbox that, if checked, will tell QLC+ to add the newly created Scene to the selected chasers. This feature is very useful when using Cue Lists in the Virtual Console panel, because the newly created Scene will appear in the Cue List for immediate use during a live performance. • **Button**: The list contains all the buttons currently present in your Virtual Console space. When selected, the buttons will be set to activate/deactivate the Scene just captured. You will see the button label chaning to "Scene from live ..." and a progress number to identify it. **Note:** Any previous function associated to the selected buttons will be overwritten ! • **Slider**: The list contains all the sliders currently present in your Virtual Console space. As with buttons, all the selected sliders will be set to control the Scene just captured.

	Note 1: A slider must be in **playback mode** to work as an intensity controller for a Scene. **Note 2:** Any previous function associated with the selected sliders will be overwritten !
Dump only non-zero values	This option will tell QLC+ to save only the channels with values that are **not** equal to zero. If you know what you're doing, this could save project space and avoid channels conflicts with other Virtual Console widgets.

Live Edit

Starting from version 4.5.0, QLC+ offers a functionality which allows the adjustment of your Functions while in Operate Mode.

The Live Edit icon is located in the QLC+ top bar, beside the DMX Dump icon and it is activated only when the user has switched to Operate Mode.
When clicking on the Live Edit icon, a Function selection panel will be displayed, allowing the selection of the function you want to adjust.
When you press OK, the correct editor will be displayed for editing that function.
At the moment, the following functions are supported for live editing:

- Scene will open a Scene Editor
  Note that by default the editor will be set in "Blind mode" when you edit a Scene that is not currently running, in case you want to make the changes effective only when the Scene is being replayed the next time it is selected. For a currently running Scene the editor will open in live mode by default.
- Chaser will open a Chaser Editor
- EFX will open an EFX Editor
- RGB Matrix will open an RGB Matrix Editor

  The types of functions listed in the Function selection can be selected by using the filter check boxes at the bottom of the panel.
  There is an option at the top of the panel to list currently running functions only.

  Note that if another function is running when performing a live edit, HTP channels in the other function may prevent some adjustments from being seen on the stage.

Fixture Manager

The fixture manager is the heart of QLC+'s fixture-oriented architecture. As its name already says, you can manage (add, remove and edit) your lighting fixtures from the fixture manager. On the left side of the manager there is a list that displays all of the fixtures in the current workspace. When a fixture is selected, the right side of the window displays the details of the selected fixture. When a Fixture Group is selected, the right side of the window is occupied by the Fixture Group Editor.

Controls

	Add new Fixture(s) to the workspace with the Add/Edit Fixture dialog.
	Add a RGB panel to the workspacewith with the Add RGB Panel dialog.
	Remove the selected fixtures from the workspace. This also removes the fixtures from ALL groups they have been assigned to.
	Edit the currently selected fixture's properties with the Add/Edit Fixture dialog.
	Opens the Channel Properties Configuration window.
	Assign the selected fixtures to a Fixture Group displayed in a popup menu. If you have no groups defined yet, you can choose to create a new one from the menu that opens when this button is clicked.
	Resign (remove) the selected fixtures from the group they are currently in. Note that removing fixtures from a group will NOT destroy the fixtures completely. Also, removal from one group will not affect the fixtures' memberships to other groups.
	Move the selected Channel Group up
	Move the selected Channel Group down
	Imports a fixture list file (.qxfl) into QLC+. Please note that fixture addresses conflicts are not handled, so it is suggested to use this functionality on an empty project.
	Exports the list of fixtures currently available on a QLC+ project into a file with extension .qxfl. This file can be used afterward with the import functionality.
	Opens the Fixtures remapping window.

Add/Edit Fixtures

The same dialog (with slight differences) is used for both adding new and editing the properties of existing Fixtures.

NOTE: When editing an existing fixture, you can always choose a completely different fixture type to replace the previous one, but if their channels don't match each other exactly, any Scene functions you may have created, will very probably do things that you don't want. Also, if the channel counts don't match, you might end up having fixtures with overlapping channels, which results also to unwanted behaviour. RGB Matrix, EFX, Chaser and Collection functions remain unaffected since they don't directly address specific channels, but are more dynamic in nature.

Controls

Fixture Model list	Shows you a list of available fixture definitions and their general types. If you don't see your favourite fixture definition on the list, you can create one yourself with the accompanying, easy to use **QLC+ Fixture Editor**. Since QLC+ is completely free software, please consider sharing the definition files with the community at https://sourceforge.net/p/qlcplus/discussion/fixtures/
Fixture Properties	• Name: The friendly name you wish to assign to the fixture • Mode: The mode you have configured to the fixture • Universe: The DMX universe where the newly-added fixture(s) are connected to • Address: The address of the (first) fixture you are adding. If you are adding multiple fixtures, each consecutive fixture will be assigned an address immediately after the previous fixture's channels, unless address gap (see below) is non-zero. • Channels: If you have selected the **Generic** dimmer device, you can define its channel count to this field. Otherwise this field is read-only and it just tells you how many channels the currently selected fixture needs in its currently selected mode. • Channel list: This field displays a more detailed list of individual channels for the currently selected fixture and its mode. For **Generic** dimmers, this field remains empty since all dimmer channels are treated as dummy **intensity** channels.
Multiple Fixtures	You can also add more than one fixture at the same time if your setup consists of multiple fixtures that are of the same make & model. These features are disabled when you are editing an existing fixture. • Amount: Number of new fixtures to create. Each newly-added fixture includes also a number after its assigned name when adding more than one fixture at a time. • Address gap: Leave this many empty channels between each new fixture

Add RGB Panel

On the market you can easily find LED strips that you can wire as you prefer to create a RGB Panel (or matrix). This dialog allows you to quickly create and setup a RGB Panel. It is a dedicated dialog to help you with the most annoying process of creating fixtures manually depending on the desiderd layout.
Please note that once the panel is created, the only way to modify the layout is manually.

Panel creation

When clicking OK on this dialog two things will happen:

- QLC+ will create a fixture for each row of the panel. They can be considered like if the panel is composed by individual RGB bars.
- QLC+ will create a fixture group representing the panel with heads already in the right displacement

Once a RGB panel is created, it is straight forward to go to the Function Manager, create a new RGB Matrix and start using the panel very quickly.

Let's have a look at every option in this panel:

Panel Properties	
Name	An arbitrary string that can be used to name the RGB panel
Universe	The universe where the RGB Panel is going to start. If the panel requires a number of channels that don't fit in a single universe, it will span across multiple universes, starting from the one selected with this option.
Address	The DMX start address where the RGB Panel has to be mapped
Size	
Columns	The RGB Panel number of columns (or the number of pixels per row)
Rows	The RGB Panel number of rows (or the number of pixels per column)
Physical	
Width	The physical width in millimeters of the RGB Panel
Height	The physical height in millimeters of the RGB Panel
Orientation	
Top-Left	The first pixel will be located on the top-left corner of the panel
Top-Right	The first pixel will be located on the top-right corner of the panel

Bottom-Left	The first pixel will be located on the bottom-left corner of the panel
Bottom-Right	The first pixel will be located on the bottom-right corner of the panel
Displacement	
Snake	The panel displacement follows a "snake" logic. It means the next pixel of the end of a row is on the same column of the next row. Then the other pixels follow in the opposite direction.
Zig-Zag	The panel displacement follows a "zig-zag" logic. It means the next pixel of the end of a row is on the first column of the next row.

Fixture Group Editor

The Fixture Group Editor is used when you select a Fixture Group in the Fixture Manager. This editor can be used to describe the physical arrangement of the fixtures and their heads that are assigned to the group.

Controls

Each Fixture Group has a name (for your convenience) and the fixture configuration in an unlimited [X,Y] grid. You can adjust the size of the grid as well as the placement of individual fixture heads on the grid.

Fixture Group Name	Change the name of the group.
	Add fixture heads to the currently selected row, filling each subsequent column on that row with the selected fixture heads. You can select individual fixture heads or complete fixtures to be added on the selected row.
	Add fixture heads to the currently selected column, filling each subsequent row on that column with the selected fixture heads. You can select individual fixture heads or complete fixtures to be added on the selected column.
	Remove the selected fixture head from the grid (and the group).
Fixture grid	The fixture grid displays the current fixture/head arrangement in the selected group. You can switch places between two heads as well as fill up spaces or create empty spaces between heads simply by dragging the heads on top of each other in the grid. When using fixture groups in RGB Matrices, each cell in the grid represents one colored (or monochrome) pixel in a graphic scene rendered by the RGB Matrix. Each cell is displayed with an info text that contains the following information: • The name of the fixture as well as an icon • **H:** represents the head number • **A:** represents the fixture's address • **U:** represents the fixture's universe
Width	Adjusts the grid width (number of columns on the X-axis).
Height	Adjusts the grid height (number of rows on the Y-axis).

Channel Groups Editor

The Channel Groups editor it activated by clicking on the "Channels Groups" tab in the Fixture Manager panel.
With this functionality (introduced in QLC+ version 4.0.0), it is possible to create groups of channels with the same functionality.
For example, if you have 20 PARs you might want to control the RED channel of all of them with a single fader.

Controls

	Add a new channels group to the workspace with the Add/Edit Channels Group dialog.
	Remove the selected channels group from the workspace.
	Edit an existing channels group in the workspace with the Add/Edit Channels Group dialog.
	Move the selected group up in the list to change the logical order.
	Move the selected group down in the list to change the logical order.

Add/Edit Channels Group

Group Name	Set/change name of the channels group
Channels list	Check channels that should be included in this channels group. Uncheck those that should not.
Apply changes to all fixtures of the same type	When checked, clicking on a channel will select/deselect the same channel on all fixtures of the same type. Use it when you want to select e.g. all Red channels of all fixture of a particular LED PAR model
External input	Select external input for this channels group for easier control in Function Manager. External input for channels groups will not work elsewhere.

Channel Properties Configuration

This window displays a tree with items nested in a Universes/Fixtures/Channels structure.
On the right side of each fixture's channel are displayed the available options that can be set to modify the behaviour of each single channel.

Channels properties

Can fade	Determine if a channel is included or excluded in the QLC+ fade transitions. By default all the channels are affected by the Fade In and Fade Out timings of the QLC+ functions. When this property is unchecked, a channel will not fade, meaning that a fade transition from 20 to 200, will set the channel immediately to 200. This is useful for example for Pan/Tilt channels of moving heads, where you want the motors to avoid fading but immediately go to the final value.
Behaviour	Force the channel behaviour according to the HTP and LTP rules. When changing the behaviour of a channel, the dropdown list will highlight in red, showing evidence that the channels has been modified. **Note: Use this functionality only if you know what you're doing and fully understand how HTP and LTP work**
Modifier	Channels modifiers are a powerful tool to modify the behaviour of a channel by acting at the end of the DMX value calculation, right before the Grand Master modification. By default all the channels will follow a linear rule, meaning that the original DMX value will be equal to the output DMX value. (0-0, 1-1, ... , 255-255) When clicking on this button, the Channel Modfier Editor panel (described below) will be displayed.

When dealing with a lot of fixtures of the same type, you might want to check the "Apply changes to fixtures of the same type" check box.
This means that changing a channel property will affect all the channels of the same type and fixture in the list.
This can save you quite a lot of time when setting up your projects for the first time.

Channel Modfier Editor

The channel modifier editor is presented like this:

On the right side of the window, a list of available modifier templates is displayed.
Clicking on a template in the list, will show the curve preview on the left side of the window.
A modifier curve is composed by a number of lines representing how a DMX channel should be modified when its value changes.
Each line starts and ends with a so called "handler". A modifier can have a virtually infinite number of handlers, but there must be always one handler on the DMX value 0 and one handler on the DMX value 255 to cover all the DMX values range.
Basically the X coordinate of the preview represents the original DMX value and the Y coordinate represents the modified DMX value.
When clicking on a handler this gets clearer as the fields above the curve preview are filled with the said values.
Handlers can be simply moved with the mouse by dragging them around or by manually changing the values of the original or modified DMX values.

QLC+ delivers a few templates that might cover the most common cases.
For example the "Invert" template can invert a Pan/Tilt channel where a product doesn't have this feature by factory default.
The "Logarithmic" template can improve the fade transitions of LED-based fixtures, trying to bring them back to a more linear effect.
To create a new template just select an existing template, give it a new name and add/remove handlers as needed with the and buttons.
When done, just click the button and your template will be saved in your user templates folder. Please refer to the Questions and Answers page to locate this folder.

Fixtures remapping

Starting from version 4.4.1, QLC+ offers a functionality called fixtures remapping.
When performing live shows in different venues, you may only be able to find out at the last minute which fixtures are installed there. Well, fixtures remapping helps you to use your existing projects in this and many other situations, such as when you need to replace a faulty fixture or when you want to use hired in equipment alongside your own.
For example, you can set up a project with just one PAR, one moving head and one scanner. When reaching the venue where the show is going to take place, you can remap your fixtures to those you find there, for example 50 PARs, 30 moving heads and 15 scanners.
With QLC+, it takes just a few minutes to do this operation !

Fixtures remapping allows you to perform 1-to-1 or 1-to-many reassignments of entire fixtures or single channels. QLC+ will try as far as possible to reassign the original channels used in the project to new channels in the same category.
When you confirm the remapping operation, a new project will be automatically saved to preserve your original project.
All the fixtures, scenes, EFX, Virtual Console widgets and audio triggers found in the original project will be remapped so that they work on the new fixtures list.

Remapping window

Let's now explain how to use the fixture remapping window, starting from a screenshot of a complete example:

And now, here is a detailed explanation of each element of the remapping window.

Source fixtures	On the left side of the window, there is a tree representing the universes, fixtures and channels used in your project. This list cannot be changed in this window.
Remapped fixtures	On the right side of the window, there is a tree representing the universes, fixtures and channels where you are going to remap your source fixtures.
	Add one or more fixtures to the Remapped fixtures list. Clicking this button will open the Add/Edit Fixtures window. Please note that once a fixture is added, it cannot be modified, so if you need specific names or quantity, remember to do it before adding the new fixtures
	Remove the selected Fixture from the remapped fixtures list.
	This is probably the most important button in the window. It allows you to determine the connection between a source ficture and a remapped fixture. The connections can be performed either between single channels or on whole fixtures. In the first case you will need to select the source channel from the source fixtures list and a target channel from the remapped fixtures list. In the second case you will need to select a fixture from the source fixtures list and a target fixture from the remapped fixtures list. Wrong selections will cause an error message to popup. For example you cannot remap a channel to a fixture and vice-versa. If the connection is valid, it will be represented as a line in the white area between the source list and the remapped list.
	Remove a previously created remapping connection. Wrong selections will cause an error popup.
Remap fixtures names	If this is checked, a new remapping connection between fixtures will also set the target fixture name be the same as the source fixture name.
Destination project name	The absolute path and name of the remapped project. For convenience, QLC+ will automatically take the original project name and will add "(remapped)" at the end of it.

Adding Fixtures

Fixture Manager

Open the fixture manager by clicking its button on the main toolbar. Alternatively you can select Fixtures from the Manager menu.

The fixture manager is the heart of QLC+ fixture-oriented architecture. As its name already says, you can manage (add, remove and edit) your lighting fixtures from the fixture manager. On the left side of the manager there is a list that contains all of the fixtures in the current workspace (currently it's empty). On the right side you can see some common information related to the currently selected fixture. On top of the fixture manager there is again another toolbar, containing the following buttons (from left to right):

Add new fixtures

Remove selected fixtures

Configure the selected fixture

Group a fixture selection

Ungroup a fixture from a group

Import a previously saved list of fixtures

Export a list of fixtures

Adding a fixture

Add a fixture to the workspace by clicking the add button.

On the left side of the dialog you can see a list of available fixture manufacturers. Each manufacturer item is actually a folder containing a number of fixture models produced by the manufacturer. You can find for example a "DJScan250" under the "Futurelight" folder. As you click a fixture from the list, you can see the Channels field on the right side change to display the number of DMX channels required by the selected fixture. There's also a list of the fixture's channels just under the Channels box.

You can edit the new fixture's name in the Name field or you can stick to the default that the application suggests. If the fixture has different operational modes (different sets of channels), you can select one from the Mode box. The fixture's DMX address can be set in the Address field and should be the same as the actual physical fixture's DMX address. The Universe field is used to assign the fixture to a physical DMX output universe. Usually each universe has its own cable coming from the computer.

If you wish to add multiple fixtures of the same type, you can increase the value in the Amount box. If you wish to leave some gap between each fixture's address space, you can change the

value in the Address gap box.

If you don't understand the DMX addressing principles, please consult your lighting equipment manuals for more information. In short, a DMX address is the first DMX channel of one fixture. In the case of a DJScan250 (which uses 6 channels), assigning for example 1 as its DMX address, reserves channels 1, 2, 3, 4, 5, and 6 to the fixture. The next fixture must then be assigned to DMX address 7 to prevent channel overlapping.

Adding a generic dimmer

Dimmers are a bit special devices, since all they can usually do is just adjust the intensities of their channels (that usually drive PAR cans and the like). If you wish to add such a device, select the Generic dimmer fixture from the list and specify the number of channels you wish the device to employ into the Channels box. Note that you cannot adjust the mode for dimmers and the channel list stays empty all the time.

Back to the fixture manager

Click OK to close the dialog and add the selected fixture(s) to the workspace.

On the left side of the fixture manager you can now see the fixture(s) that you just added. On the right side, you can see information on the currently selected fixture. You can edit the fixture's name, address and universe by clicking the Configure button. You can also change the fixture definition thru the configuration dialog.

Function Manager

With the function manager you can manage all of your functions that do the actual work of automating your lights. You can create new functions, remove and edit existing ones as well as create copies of them. Controls are on the upper part of the window; the lower part displays all of your functions as well as the function editor for the currently selected function (if any).

Each function type can be found under its own category: Scene for scenes, EFX for EFX's etc. When a new function is created or an existing one selected, the appropriate editor is chosen and displayed on the right hand side of the Function Manager window. Changes made in the editor pane are stored immediately in the functions themselves and no additional OK clicks are required.

Controls

	Create a new Scene and edit it using the Scene Editor.
	Create a new Chaser and edit it using the Chaser Editor.
	Create a new Sequence and edit it using the Chaser Editor.
	Create a new EFX and edit it using the EFX Editor.
	Create a new Collection and edit it using the Collection Editor.
	Create a new RGB Matrix and edit it using the RGB Matrix Editor.
	Create a new RGB Script and edit it using the RGB Script Editor.
	Create a new Audio function to be used in Chasers or Shows.
	Create a new folder in the selected category. A folder can be renamed by double clicking on it. When a folder is selected, a newly created function will be added to it. To move existing functions inside a folder, just select and drag them into it. In the same way they can be moved outside a folder by dragging them onto a category item. When deleting a folder, all the functions and subfolders contained into it will be deleted as well. **Note:** Empty folders will not be saved into your project.
	Open the Function selection dialog to choose a startup function. The selected function will be started every time QLC+ switches to operate mode. This is very useful when running QLC+ in kiosk mode (thus using -k or -p flags) A special entry named "<No function>" is present to remove a previously set startup function.

	Start the Function Wizard.
	Create a copy of each of the selected functions.
	Permanently remove the selected function(s).

Scene Editor

The scene editor, as its name suggests, is used to edit Scene functions. The editor is divided into tabs; with the first, **General** tab, you control the list of fixtures and channel groups that take part in the scene editing, together with the Scene name.
All subsequent tabs are used to control the individual channel values for each fixture and, if any are defined, the channels groups values.

General Tab Controls

On the left hand part of the screen, the buttons to control the fixtures used in the scene are displayed.

Scene name	Change the name of the scene.
	Add an existing Fixture to the scene.
	Remove the selected Fixture(s) from the scene.
	Enable all channels of the selected fixtures.
	Disable all channels of the selected fixtures.

On the right hand part of the screen, the buttons to control the channel groups used in the scene are displayed

	Enable all the selected channel groups.
	Disable all the selected channel groups.

Hint: It's useful to know that when a channels group is checked/unchecked, the fixtures controlled by the group will be automatically added to the left panel. The channels controlled by the group will also be automatically enabled/disabled on each fixture.

Channel Groups Tabs

This tab will be displayed only if one or more channel groups are selected in the General tab. Each Channel Group will be displayed with a quick macro access button (Click And Go if supported), a label with the group value, a fader and the name of the group.

Fixture Tabs

Each fixture is represented by its own tab that contains sliders for each of the fixture's channels.

Each channel can be enabled or disabled with a check box at the top of the channel unit. The value of each channel can be set either by typing the value to the edit box at the top of the slider or by moving the slider. Channels that provide multiple functions such as gobos, colors, etc. also have a button above the channel slider - this button can be used to directly select a specific function or capability provided by that channel.
Hint: The keyboard shortcut to move between channels values edit boxes is '**Tab**' to move to the right and '**Shift + Tab**' to move to the left.

Channel enabled/disabled status

If a channel has not been enabled, the scene will not touch that particular channel's value, ever.

If a channel has been enabled, the scene will change that channel's value to the value that is defined in the scene. This is useful, for example, when you wish to control only the dimmer channel of a scanner fixture - you wouldn't want the scene to touch the scanner's pan, tilt, color or gobo channels when you just want to fade in or fade out with the dimmer channel.

Controls

	Enable all channels from the current fixture. In all channels mode, select channels for all fixtures.
	Disable all channels from the current fixture. In all channels mode, disable channels for all fixtures.
	Go to the previous tab in the view. If the first tab is selected, this will go to the last tab in the view. Keyboard shortcut: '**ALT+Left**'
	Go to the next tab in the view. If the last tab is selected, this will go to the first tab in the view. Keyboard shortcut: '**ALT+Right**'
	The Copy functionality has 2 modes: • Copy the values (and enabled/disabled states) of all channels in the current fixture to the clipboard. • Copy only the selected channels (CTRL-clicked, yellow background) values into the clipboard.
	Paste the values from clipboard to the current fixture. If no channels were selected during 'copy', then all the enabled/disabled states will be pasted along with the channel values.
	The 'Copy all' functionality has 2 modes: • Copy all values (and enabled/disabled states) of all channels to all other fixtures taking part in the scene. • Copy only the selected channels (CTRL-clicked, yellow background) values to all the other fixtures taking part in the scene.
	Launch a color tool to select a specific color and set that color to the current fixture; This

	feature is enabled only for fixtures that are capable of CMY/RGB color mixing.
	Launch a position tool (similar to XYPad in Virtual console) to select a head/mirror position for the current fixture; This feature is enabled only for fixtures that have either pan or tilt channels. All heads of current fixture will be changed to the same value.
	Show/Hide the Speed Dial widget, used to adjust the Scene parameters such as Fade In and Fade Out
	Switch between "tab view" and "all channels view". The first mode will display one tab for each fixture, while the second will display a single tab for all the fixtures
	Toggle blind mode for the selected fixture.
	Clone the current scene and add a new step to the Chaser selected from the drop down list beside this button

Chaser Editor

The chaser editor, as its name suggests, is used to edit Chaser functions.

A Chaser is composed by steps and each step is represented by

- A progressive number
- The related function name
- Fade in time
- Hold time
- Fade out time
- Step total duration time
- A note field in case you need to write something to remember what the step does

The timings steps and the note field can be modified by double clicking on them. By default the times are taken in seconds (for example 135 means 2m15s), unless you use the same syntax QLC+ uses (for example 1m12s.80)

Controls

Chaser name	Change the name of the chaser.
	Show/Hide the Speed Dial widget, used to facilitate the Chaser parameters tuning like Fade In, Fade Out and Hold
	Start to run the Chaser to test the playback beginning from the selected step
	Stop a previously started playback
	When playback is active, go to the previous step in the Chaser
	When playback is active, go to the next step in the Chaser
	Copy the selected steps into the clipboard.
	Paste the previously copied steps from clipboard to the current position selected. If none, the steps will be appended at the end of the list
	Remove the selected steps from the list and copy them into the clipboard.
	Add an existing Function to the chaser, using the Select Function dialog. The function are inserted after the currently selected step.
	Remove the selected step from the chaser.

	Move the selected step above/before the previous step.
	Move the selected step below/after the next step.
Run Order	• Loop: Run through the steps over and over again. • Single Shot: Run through the steps once and then stop. • Ping Pong: Run through the steps over and over again, reversing direction at both ends. • Random: Run through the steps over and over again in random order.
Direction	• Forward: Run through the steps from start to end; 1, 2, 3... • Backward: Run through the steps from end to start; ...3, 2, 1
Fade In Speed	• Common: Apply the same speed for all the steps in this Chaser • Per step: Apply a user defined speed for each step in this Chaser • Default: Apply the default speed for each step (0 seconds).
Fade Out Speed	• Common: Apply the same speed for all the steps in this Chaser • Per step: Apply a user defined speed for each step in this Chaser • Default: Apply the default speed for each step (0 seconds).
Step Duration	• Common: Apply the same duration for all the steps in this Chaser • Per step: Apply a user defined duration for each step in this Chaser

Show Editor

The Show Editor is a panel to display the current structure of a Show created with the Show Manager.
At the moment the Show Editor can only rename a Show, which is not possible in the Show Manager.

The tree view of this panel shows useful information regarding the displayed Show such as:

- **Function:** the function name
- **Steps:** the number of steps that compose a Sequence
- **Start Time:** the time at which a sequence is going to be reproduced
- **Duration:** the duration of the sequence

EFX Editor

The EFX editor, as its name suggests, is used to edit EFX functions. The view is split into two tabs:

- **General** tab is for selecting Fixtures, speed and fixture order.
- **Movement** tab is for selecting details on how the fixtures should move their beams.

General Tab Controls

	Adjust the EFX speed settings: • **Fade In:** The time it takes for the fixtures in the EFX to fade their intensity channels up • **Fade Out:** The time it takes for the fixtures in the EFX to fade their intensity channels back to zero • **Duration:** The duration of one full round using the selected pattern
	Enable the output and run the EFX to test how it looks like
EFX name	Change the name of the EFX.

Fixture list	• Step: Shows the order in which the fixtures start their movement in serial/asymmetric order • Fixture: The name of each fixture that has been added to the EFX • Reverse: When checked, the fixture reverses its movement as opposed to non-reversed fixtures. The "normal" direction is set by direction in the "Movement" page. • Start Offset: Value in degrees, where on the movement path this fixture should start • Intensity: The value set to the fixture's intensity/dimmer channel when the fixture starts to move
	Add a Fixture to the EFX, after the currently selected position. Since EFX is used for moving the head or mirror of an intelligent fixture, only fixtures with **pan and tilt** capability (8bit or 16bit) can be added to an EFX.
	Remove the selected fixtures from the EFX.
	Raise the selected fixture upwards (earlier) in serial order.
	Lower the selected fixture downwards (later) in serial order.
Fixture order	Change the EFX's fixture order—see below.

Movement Tab Controls

Pattern area	Shows a 2D-projection of the fixtures' head/mirror movement. Every time you change a parameter, one small dot for each fixture travels the complete path in its selected direction from its start offset. The speed of the movement reflects selected speed settings.
Direction	Default direction of the fixtures (can be altered individually for each fixture by the checkbox in the **Reverse** column) • Forward: The fixtures move forwards along the pattern path • Backward: The fixtures move backwards along the pattern path
Run order	• Loop: Run thru the steps over and over again. • Single Shot: Run thru the steps once and then stop. • Ping Pong: Run thru the steps over and over again, reversing direction at both ends.
Pattern	Select the movement pattern algorithm. • Circle • Eight • Line: goes from one end to the other and back; faster in the middle, slower at the ends • Line2: goes in one direction only; speed is always the same • Diamond

	• Square • SquareChoppy • Leaf • Lissajous
Parameters	• Width: Choose the pan width (0-255) • Height: Choose the tilt height (0-255) • X Offset: Move the pattern's horizontal (pan) centerpoint (0-255) • Y Offset: Move the pattern's vertical (tilt) centerpoint (0-255) • Rotation: Rotate the pattern along its axis (0-360 degrees) • Start Offset: Where along the path the movement should start (0-360 degrees) • X Frequency: Change the **Lissajous** pattern's X (horizontal) frequency (0-32) • Y Frequency: Change the **Lissajous** pattern's Y (vertical) frequency (0-32) • X Phase: Change the **Lissajous** pattern's X (horizontal) phase (0-360 degrees) • Y Phase: Change the **Lissajous** pattern's Y (vertical) phase (0-360 degrees)
Relative	See below.

Fixture Order

Fixtures taking part in an EFX function can be set to follow the algorithm in certain order:

- **Parallel:** all fixtures follow the same pattern synchronously
- **Serial:** fixtures start following the pattern one after the other, with a little delay between each of them.
- **Asymmetric:** all fixtures start moving simultaneously, but with similar offset as in the Serial mode.

Direction

EFX functions' direction can be reversed for all fixtures at once or on a per-fixture basis. The function can also be set to do an infinite loop, an infinite ping-pong-loop (direction is reversed after each pass) or it can run through just once, in a single-shot mode, after which it terminates by itself. If the function is set to do an infinite loop, it must be stopped manually.

EFX Intensity

When EFX is started, it will set the intensity of all included fixtures to the preset value. That usually means it set them full on. If you don't want this, just set the preset intensity to something lower, perhaps 0. That way you can control intensity of the fixture using other means like VC sliders for the intensity/dimmer channels, or by another chaser.

Relative Mode

EFX position is absolute by default-in other words, the selected EFX will exclusively control the X/Y position of the specified heads. When the Relative Mode checkbox is enabled, the EFX position acts as a layer on top of any position that has already been set (e.g. by a scene or even another EFX). In other words, the EFX is relative to current fixture position.

In absolute mode. the EFX can be set to run at specific head position (e. g. do circle downstage center, stage left, etc.)

In relative mode, the center of the preview window (offset x=127, offset y=127) will be applied to current head position.

It is useful to reduce number of EFX presets: let's say we want to have 3 types of EFX (pan saw, tilt saw, circle) at 4 places (e.g. 4 stage corners). In absolute mode that means 3x4 = 12 presets. If we want to change something, we have to edit many functions. We most probably need also one VC button for each. In relative mode, we create one EFX preset for each EFX type (pan saw, tilt saw, circle), and we set offset to neutral (x=127, y=127). Then, we create scenes with PAN & TILT channels for each position. Now we have only 3+4 presets (and 3+4 VC buttons, preferably in 2 solo frames).

Tips&Tricks

In relative mode, it is also possible to fade between positions (set fade time in scenes) while the EFX is running.

Using XYPad and relative mode, it is possible to move EFX to any place during the show.

Collection Editor

The collection editor, as its name suggests, is used to edit Collection functions.

Note: Collections don't have speed setting; each function you include in a collection follows its own speed settings.

Controls

Collection name	Change the name of the collection.
	Add an existing Function to the collection, using the Select Function dialog. The order of the functions has no significance.
	Remove the selected functions from the collection.

RGB Matrix Editor

The RGB matrix editor, as its name suggests, is used to edit RGB matrix functions. The function works on predefined Fixture Groups created by the user with the Fixture Manager.

Controls

RGB matrix name	Change the name of the RGB matrix.
	Adjust the RGB Matrix speed settings: • **Fade In:** The time it takes for the fixtures in the RGB Matrix fixture group to fade their intensity channels up • **Fade Out:** The time it takes for the fixtures in the RGB Matrix fixture group to fade their intensity channels back to zero • **Duration:** The duration of each step of the selected pattern
	Convert the current RGB Matrix into a Sequence. This is useful to speed up the creation of a Show. Please note that if the selected pattern generates random data, this functionality will generate a different Sequence every time.
	Switch the preview mode between circles and squares
	Make the RGB Matrix run as if it were started from a Virtual Console button. Note that the preview stops while the function is running.
Fixture group	The Fixture Group that is controlled by this RGB matrix.
Preview area	Shows a preview of the currently selected pattern imposed on the fixtures defined in the currently selected fixture group. Note that the preview doesn't show the difference between RGB-capable and monochrome/fixed color fixtures.
Pattern	Select the pattern and color that are used on the selected fixture group for drawing graphics.
Plain Color	When the **Plain Color** pattern is selected, all the pixels of the RGB Matrix will be lit to the color selected
Animated	Available only when **Text** is the selected pattern. • The text edit field is used to edit the text content that is scrolled/flashed on the matrix.

Animated Text	• The font button is used to select the font (tip: bitmap fonts work best) • The drop-down box is used to select the animation style (Horizontal, Vertical or Letters)
Audio Spectrum	When the **Audio Spectrum** pattern is selected, QLC+ will start capturing data from the chosen audio input device and will represent the audio spectrum as vertical bars in the RGB Matrix with the selected start and end color. For an optimal usage of this pattern, the hold time of the Matrix must be set to a reasonably low value (e.g. .20)
Image	Available only when **Image** is the selected pattern. • The text edit field is used to edit the image filename • The image button is used to select the image • The drop-down box is used to select the animation style (Static, Horizontal, Vertical or Animation) Supported formats: PNG, XPM, JPG, GIF (for animated gifs only first frame is used). Styles: • **Static** - the image is statically displayed on the matrix. If the image is smaller than the matrix, it is repeated in both directions. (Tip: for stripes, use 1-point-high images) • **Horizontal** - the image is scrolled horizontaly. Again, image is repeated in both directions. • **Vertical** - same as **Horizontal**, but in vertical direction. • **Animation** - play animation. Stack the frames (of width equal to matrix's) side by side in one image. For example, when matrix is 8x8 and you want to have 4 frames, make image 32x8. The image is still repeated vertically.
Offset	Available only when **Text** or **Image** is the selected pattern. • X: Used to shift the pattern along the horizontal x-axis (negative values go to the left, positive values go to the right) • Y: Used to shift the pattern along the vertical y-axis (negative values go up, positive values go down)
Run Order	• Loop: Run through the steps over and over again. • Single Shot: Run through the steps once and then stop. • Ping Pong: Run through the steps over and over again, reversing direction at both ends.
Direction	• Forward: Run through the steps from start to end; 1, 2, 3... • Backward: Run through the steps from end to start; ...3, 2, 1

RGB Script Fundamentals

Users can write their own RGB scripts to produce custom graphics projection with the RGB Matrix function. The scripts' operation principle is to produce a number of RGB maps that each represent one step in the graphics animation. The principle is basically the same as in movies: the audience observes a moving picture, which in reality is only a sequentially-played stream of static images that produce an illusion of movement.

The scripts themselves are written in ECMAScript, which is also known as JavaScript. Note that the language is case-sensitive and as a de-facto standard follows camel case rules, so make sure you write everything correctly and pay special attention to the required API features.

Script files should be named after the script's name and must have a **.js** extension. Depending on platform, the files should be placed either in the QLC+ system script directory or, preferably, the user script directory:

- **Linux user dir:** ~/.qlcplus/rgbscripts/
- **Linux system dir:** /usr/share/qlcplus/rgbscripts/

- **OSX user dir:** ~/Library/Application Support/QLC+/RGBScripts
- **OSX system dir:** /Applications/QLC+.app/Contents/Resources/RGBScripts

- **Windows user dir:** Not applicable
- **Windows system dir:** C:\QLC+\RGBScripts

RGB Script API

Foundation

The scripts must be self-executing, i.e. when they are evaluated, the script itself is put inside an anonymous function that executes itself and returns an object that contains the required API functions:

```
(
function() { // Anonymous function starts here
    var algo = new Object;
    return algo; // Return the script object
} // Anonymous function ends here
)() // Anonymous function is executed here upon evaluation
```

Properties

However, a script with nothing more than an empty object does nothing, no matter how self-executing it might be. You must also declare some **properties** for the returned object so that QLC+ knows how to use the script and to show it to the user (you). So, you need to declare the following properties for the returned script object:

- **apiVersion:** The API version that the script follows. Currently, the accepted values are '1' or '2'.
  apiVersion 1 allows simple scripting and easier coding, while apiVersion 2 offers advanced features described below.
  Any other value will cause the script to be treated as invalid.
- **name:** The name of your script. This name appears in the pattern selection box in the RGB Matrix Editor
- **author:** The name of the person who has written the script. **You.**
- **acceptColors (optional):** Informs QLC+ if the script can accept a start color, an end color or none.
  'acceptColors = 0' means no colors are accepted/needed. It means that the script will autonomously generate the colors information for the matrix pixels.
  'acceptColors = 1' means only the start color is needed by the script.
  'acceptColors = 2' means both start and end colors will be accepted by the script.
  If 'acceptColors' is not provided, QLC+ will default to value '2'.

With this in mind we add declarations for these three properties to the script:

```
(
function() {
    var algo = new Object;
    algo.apiVersion = 2; // Can be '1' or '2'
    algo.name = "My cool RGB script";
    algo.author = "Your name";
    algo.acceptColors = 2; // Can be '0', '1' or '2'
    return algo;
}
)()
```

Functions

Now we are getting to the actual business of producing data for the RGB Matrix. The current API version uses two functions to achieve this:

- rgbMapStepCount(width, height)
- rgbMap(width, height, rgb, step)

No assumptions on the calling order or argument immutability should be made, i.e. do not cache the values from either function and assume that they are valid until the end of the world. The parameters might change at any point (usually when the user changes the matrix size, color or direction) thus invalidating any cached values.

rgbMapStepCount(width, height)

When QLC+ calls this function, it wants to know the number of different RGB maps returned by the rgbMap() function, when the RGB matrix size is **width** times **height** pixels. It must always return the same result when called with the same **width** and **height** parameters again and the result must not change over time.

Parameters:

- **width:** The width of the grid
- **height:** The height of the grid
- **Return value:** Number of steps produced by rgbMap() with the given width and height parameters

So, now we add this support function to the script:

```
(
function() {
    var algo = new Object;
    algo.apiVersion = 1;
    algo.name = "My cool RGB script";
    algo.author = "Your name";

    algo.rgbMapStepCount = function(width, height) {
        ...
        return number_of_steps_when_width_is_oranges_and_height_is_jabberwocky;
    }

    return algo;
}
)()
```

rgbMap(width, height, rgb, step)

This function is the actual brains of the script. It produces two-dimensional arrays whose size MUST be **height** times **width**. I.e. the array returned from this function must contain **height** items and each of these items must be an array that contains **width** items that must be 32bit integers, representing an RGB color as understood by QRgb without alpha channel (0x00RRGGBB). The **rgb** parameter is an integer-representation of the color selected by user in the RGB Matrix Editor. The **step** parameter tells the step number requested by the RGB Matrix function and is guaranteed to be within (0, rgbMapStepCount(w, h) - 1).

Parameters:

- **width:** The width of the grid
- **height:** The height of the grid
- **rgb:** The color selected by user
- **step:** Current step number to produce (between 0 and rgbMapStepCount(w, h) - 1)
- **Return value:** An array containing [height] arrays each containing [width] integers

Just like the previous function, we also add this other one to the script. Now we have a full and ready template for any RGB script for your indulgence.

```
(
function() {
    var algo = new Object;
    algo.apiVersion = 1;
    algo.name = "My cool RGB script";
    algo.author = "Your name";

    algo.rgbMapStepCount = function(width, height) {
        ...
        return number_of_steps_when_width_is_oranges_and_height_is_jabberwock;
    }

    algo.rgbMap = function(width, height, rgb, step) {
        ...
        return a_2d_array_of_arrays;
    }

    return algo;
}
)()
```

API version 2

RGB Script API version 2 introduces the concept of **Properties**. With properties, a Script can interact with the QLC+ engine by exchanging parameters, thus extending the possibilities of a RGB Script.
Examples of such properties can be the animation orientation, the number of objects to be rendered, etc..

Adding properties to your script is fairly easy, but requires a specific syntax explained in this paragraph.
Let's make an example:

```
algo.orientation = 0;
algo.properties = new Array();
algo.properties.push("name:orientation|type:list|display:Orientation|values:Horizontal,Vertical|write:setOrientation|read:getOrientation");
```

The three lines above specifies the following:

1. The script has an internal property represented by the variable 'orientation'
2. Create the properties array. This is needed just once
3. Push (add) the property description into the properties array. This is what the QLC+ engine will actually read to exchange data with the script

The third line is indeeed the most interesting and the syntax of the string stored in the properties array must be followed to avoid errors during the script loading.
Attributes must be in the form '**name:value**' and each attribute must be separated from the others by a pipe '|' character.
Following a table of the accepted attributes and the meaning of their values.

Attribute name	Attribute value
name	The name used by QLC+ to uniquely identify a property. The same name must not be used twice in a script. It is a good practice to use the same name of the 'algo' variable that actually stores the property.
type	This is a fundamental attribute to allow QLC+ to correctly handle a property. The 'type' attribute must be placed before the 'values' attribute. Accepted values are: • **list**: defines a list of strings that will be displayed by the QLC+ RGB Matrix Editor • **range**: defined a range of integer values that this property can handle • **integer**: an integer value that QLC+ can exchange with the script • **string**: a string that QLC+ can exchange with the script The type attribute implicitely defines also how the QLC+ RGB Matrix editor will display the property. For example a 'list' is displayed like a drop down box, a 'range' and an 'integer' are displayed as a spin box, a 'string' is displayed as an editable text box.
display	An string that QLC+ will display in the RGB Matrix editor for the user. It is a good practice to use a meaningful and human readable string to allow users to immediately understand what a property does
values	This attribute can be applied only when type is 'list' or 'range'. It defines the possible values that the property can assume. A 'list' type will look like 'one,two,three' while a 'range' type will look like '2,10'. Values must be separated by a comma ',' character. A 'range' type cannot have more than two values.

write	Defines the name of the Script function that QLC+ should call to write the property value.
	In this function the Script's writer should implement all the necessary actions to apply the property change.
	The write method of the example above is the following:
	`algo.setOrientation = function(_orientation) {`
	`if (_orientation == "Vertical")`
	`algo.orientation = 1;`
	`else`
	`algo.orientation = 0;`
	`}`

read	Defines the name of the Script function that QLC+ should call to read the property value.
	The read method of the example above is the following:
	`algo.getOrientation = function() {`
	`if (algo.orientation == 1)`
	`return "Vertical";`
	`else`
	`return "Horizontal";`
	`}`

Development Tool

There is a development tool available in the QLC+ source repository that makes it easier to debug and test your custom scripts with a web browser. To use the tool, you must download the following two files to a directory on your hard disk, open the **devtool.html** file with your browser and follow its instructions:

- devtool.html
- devtool.js

(Right-click and "Copy Link Location" works probably best)

Example Script: Full Columns

```
/*
  Q Light Controller Plus
  fullcolumns.js

  Copyright (c) Heikki Junnila

  Licensed under the Apache License, Version 2.0 (the "License");
  you may not use this file except in compliance with the License.
  You may obtain a copy of the License at

      http://www.apache.org/licenses/LICENSE-2.0.txt

  Unless required by applicable law or agreed to in writing, software
  distributed under the License is distributed on an "AS IS" BASIS,
  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
  See the License for the specific language governing permissions and
  limitations under the License.
*/

(

/**
 * This algorithm produces fully-lit columns, meaning all pixels on a single
 * column are lit together.
 */
function()
{
    var algo = new Object;
    algo.apiVersion = 1;
    algo.name = "Full Columns";
    algo.author = "Heikki Junnila";

    /**
     * The actual "algorithm" for this RGB script. Produces a map of
     * size($width, $height) each time it is called.
     *
     * @param step The step number that is requested (0 to (algo.rgbMapStepCount - 1))
     * @param rgb Tells the color requested by user in the UI.
     * @return A two-dimensional array[height][width].
     */
    algo.rgbMap = function(width, height, rgb, step)
    {
        var map = new Array(height);
        for (var y = 0; y < height; y++)
        {
            map[y] = new Array();
            for (var x = 0; x < width; x++)
            {
                if (x == step)
                    map[y][x] = rgb;
                else
                    map[y][x] = 0;
            }
        }

        return map;
    }

    /**
     * Tells RGB Matrix how many steps this algorithm produces with size($width, $height)
     *
     * @param width The width of the map
     * @param height The height of the map
     * @return Number of steps required for a map of size($width, $height)
     */
    algo.rgbMapStepCount = function(width, height)
    {
        // Each column is lit completely at a time, so because there are $width
        // columns in the map, the number of steps must be $width to light all
        // columns per round.
        return width;
    }

    return algo;
}
)()
```

Script Editor

The Script Editor, as its name suggests, is used to edit Script functions.

It basically consists in a text editor where the user can write a script using a meta-language with a syntax described below.
The script can be manually modified if you have a full understanding of the language syntax, otherwise a few helper buttons are available on the rightmost side of the editor to speed up and facilitate the script creation/editing.
Each line of a script will be executed by QLC+ in a sequential order.

Controls

	Enable the output and run the Script to test its execution
Script name	Change the name of the Script.
	When clicked, this button will display a popup where you can choose the script snippet you want to insert at the current cursor position in the editor. • **Start Function**: Opens the Functon Selection dialog to select a Function to be started. This operation will automatically add a comment at the end of the line of code, with the QLC+ name of the selected Function. • **Stop Function**: Opens the Functon Selection dialog to select a Function to be stopped. If at the moment of execution of this line of code the selected Function is not running, this instruction will have no effect. This operation will automatically add a comment at the end of the line of code, with the QLC+ name of the selected Function. • **Set Fixture**: Opens the Fixture Channels selection dialog, where you can choose the channels you want to control with the Script. If multiple channels are selected, this operation will add one line of code for each selected channel. By default, the DMX value generated by this opration will be 0 and has to be changed manually with the script text editor. This operation will automatically add a comment at the end of the line of code, with the QLC+ name of the selected fixtures and channels. • **System Command**: Opens a file dialog to choose the external program/script to run. Please note that the selected file must be executable to be accepted. Once a file has been selected another dialog will ask to enter the program/script arguments. If none are required, just leave the field empty. • **Wait**: Opens a dialog requesting the time to be waited before the script can execute the next instruction • **Comment**: Opens a dialog requesting the text of the comment to insert at the cursor position in the editor. A comment has a C language style appearance: it starts with "//" and everything until the end of the line is then considered a comment • **Random Number**: Opens a dialog requesting the range of values where to perform a randomization of a number. The resulting code snippet will be inserted at the editor current cursor position. This can be useful to randomize DMX values set through the setfixture keyword. See below. • **File Path**: Open a file dialog to select a file. The absolute path and the file name will be insterted at the editor current position. If a path contains spaces, it will be writen within quotes
	Cut the currently selected text in the editor for a later paste
	Copy the currently selected text in the editor for a later paste
	Paste a previously cut or copied text at cursor position in the editor
	Undo the last performed operation on the editor
	Check the Script syntax. A popup message will be displayed indicating if the Script is OK or the line numbers where syntax errors have been found.

Language syntax

The QLC+ Script meta-language is based on a **keyword:value** model, with some insertions of C language syntax rules.
Each line of code is parsed by the QLC+ engine and verified to detect the presence of syntax errors.
Here's a table describing each keyword accepted by the Script engine and its syntax.

startfunction	**Type**: keyword **Description**: starts a QLC+ Function with the given ID. **Syntax**: startfunction:functionID *functionID* is an integer number of the ID assigned by QLC+ to a Function. Since IDs are not exposed to QLC+ users, in this case it is convenient to use the helper button on the rightmost side of the editor, which also add a comment with the Function name. Eventually a user will learn the ID of a Function and therefore use it to manually add more code to the script. **Example**: `startfunction:2 // Green scene`
stopfunction	**Type**: keyword **Description**: stops a running QLC+ Function with the given ID. **Syntax**: stopfunction:functionID *functionID* is an integer number of the ID assigned by QLC+ to a Function. See *startfunction* description. **Example**: `stopfunction:0 // Blue scene`

systemcommand	**Type**: keyword
	Description: execute a program or a script at the provided absolute path with the (optional) provided arguments.
	Syntax: systemcommand:programPath arg:arg1 arg:arg2 ... arg:argN
	
	programPath is the absolute path of an executable program or script. For example "/usr/bin/vlc" or "C:\Tools\myTool.exe"
	If the path to an executable contains spaces, it must be written between quotes.
	arg1 ... argN are the arguments to be used when executing *programPath*. If no arguments are needed, then the arg: keywords are not necessary.
	If an argument contans spaces it must be written between quotes.
	Examples:
	`systemcommand:/usr/bin/vlc arg:-f arg:/home/user/video.mp4 // plays my video with VLC in fullscreen`
	`systemcommand:"C:\Program Files\Tools\My Tool.exe" arg:"D:\My Files\My file.txt"`

setfixture	**Type**: keyword **Description**: sets a QLC+ Fixture channel to the provided DMX value. **Syntax**: setfixture:fixtureID ch:channelIndex val:DMXValue *fixtureID* is an integer number of the ID assigned by QLC+ to a Function. Since IDs are not exposed to QLC+ users, in this case it is convenient to use the helper button on the rightmost side of the editor, which also add a comment with the fixture and channel name. Eventually a user will learn the ID of a fixture and the index of a channel and therefore use them to manually add more code to the script. *channelIndex* is an integer number representing the fixture channel number. Channels indices here start from 0. *DMXValue* is the actual DMX value to be set to the specified fixture channel. It ranges from 0 to 255 **Example**: `setfixture:0 ch:1 val:135 // Generic RGB, Red. Sets the red channel of a Generic RGB fixture to DMX value 135`

wait	**Type**: keyword
	Description: wait for the provided amount of time before executing the next line of code.
	Note that a wait time can be randomized too, following the *random* syntax described below.
	Syntax: wait:time
	
	time can be either an integer number of milliseconds or a string representing the wait time in the QLC+ way: **h**m**s.**
	Examples:
	`wait:1800 // Waits for 1 second and 800 milliseconds`
	`wait:03s.20 // Waits for 3 seconds and 200 milliseconds`

comments	**Type**: Helper macro **Description**: comments can be inserted at any position in the script code and they do not affect the script execution. They are normally used to give a meaning to a line of code. QLC+ Scripts comment follow the C Language style rule of the "//" syntax. Basically everything written after "//" will be considered a comment until the end of the line of code. So pay a particular attention to not writing meaningful code after a "//" and expect it to be run, cause it won't. Comments can be added at the end of a Script line of code or they can take a whole line, for example to describe an entire block of code.

random	**Type**: Helper macro
	Description: generates a random integer number between the provided minimum and maximum values
	Syntax: random(min,max)
	
	min is the minimum value the randomization can reach. It can be either an integer number or a time string
	max is the maximum value the randomization can reach. It can be either an integer number or a time string
	Examples:
	`wait:random(02s.00,05s.00) // Waits a random time between 2 and 5 seconds`
	
	`// set channel 3 of fixture with ID:1 to a random DMX value between 20 and 235`
	`setfixture:1 ch:2 val:random(20,235)`

Function Wizard

The Function Wizard's purpose is to help users create some common functions for a quick start.

Controls

Fixture List	List of fixtures that will be included in the functions created with the wizard. You can see each fixture's capabilities for Color-, Gobo- and Shutter- type scenes in the **Supported capabilities** column.
	Include Fixtures to the functions that will be created with the wizard.
	Remove the selected fixtures from the function to be created.
Scenes	• Colors: Create scenes for changing the color simultaneously on all currently included fixtures. For this to work, the fixtures must have at least one color channel that controls a fixed-color wheel. • Gobos: Create scenes for changing the gobo simultaneously on all currently included fixtures. For this to work, the fixtures must have at least one gobo channel. • Shutter: Create scenes for changing the shutter position simultaneously on all currently included fixtures. For this to work, the fixtures must have at least one shutter channel.

Select Function(s)

Whenever a Function needs to be selected, for example when you're adding steps to a Chaser, using the Chaser Editor, the **Select Function** dialog is used.

Sometimes the destination that you are selecting the function for might allow the selection of multiple functions, for example, when adding steps to a Chaser or functions to a Collection. In these cases, the order in which the functions are selected will also be the order in which they are added to their destination. On the other hand, when attaching a function to a button in the Virtual Console, you may select only one function at a time.

Refer to your operating system manual for documentation on how to select multiple items in a list.

Controls

Functions to display	There are two radio buttons on the top of the window: **All functions** and **Running functions**. By clicking on them, the list below will be filled only with the desired functions. For example, when performing a Live edit, you might want to display only the running functions. Obviously, in design mode, selecting "Running functions" will show an empty list.
Function list	• Name: The names of available functions • Type: The type of each of the available functions
Filter	You can filter the function list so that only those function **types** that have been checked are shown. Filtering is especially useful if you have lots of functions to select from. • Scenes: Display Scene functions in the selection list • Chasers: Display Chaser functions in the selection list • EFX's: Display EFX functions in the selection list • Collections: Display Collection functions in the selection list • RGB Matrices: Display RGB matrix functions in the selection list • Show: Display Show functions in the selection list • Audio: Display Audio functions in the selection list

Select Fixture(s)

Whenever a Fixture needs to be selected, for example when adding fixtures to a Scene function, the **Select Fixture(s)** dialog is used.

The dialog is very straightforward; there is only a list of fixtures that you can select from. In some cases you are allowed to select multiple fixtures, for example, when adding fixtures to a Scene or an EFX. In these cases, the order in which the fixtures are selected will also be the order in which they are added to their destination.

Refer to your operating system manual for documentation on how to select multiple items in a list.

Show Manager

The Show Manager has been introduced in QLC+ starting from version 4.0.0, after forking the original QLC code on November 5th, 2012.

This feature is meant to give users the possibility of setting up a time driven show in a friendly and completely graphical way.
The graphic interface shows a multitrack view, typical of audio sequencers or video editing softwares, and with it users are allowed to place QLC+ Functions at the desired place and time in the view.
Show Manager gives a lot of flexibility during a Show creation thanks to its visual-oriented approach. Once understood the basic elements, it is very easy to create, move or edit the existing functions and improve a Show by adding new tracks to it.

Typical use cases of Shows are those gigs where a band plays songs following a metronome and the light show has always to be the same, following the music.
Another case are visual entertainment shows, where dancers or singers follow some music and lights create the atmosphere at the right time.

The Show Manager drives users to make large use of the Sequence function. Here's the explanation of the difference between a Chaser and a Sequence.

Sequences vs Chasers

Even if the Sequence and the Chaser functions have some common ground, they're not the same thing.
If not done yet, you are invited to read once again their definitions in the Basic Concepts & Glossary page of this documentation.
The main differences are:

- **Steps**: The steps of a Chaser can represent any QLC+ Function, while the steps of a Sequence represents different values of the same Scene.
  In other words, a Chaser is an independent function, while a Sequence can exist only on top of a Scene.
  The reason for this is, as mentioned before, the visual approach of the Show Manager. If a track of a Show is the graphical representation of a Scene, then it's more intuitive to think that each Sequence created on that track is a function controlling the values of that Scene.
- **Order**: Chasers can be reproduced in any order (Forward, Backward, Ping-Pong, Random) while in the Show Manager, Sequences are always reproduced from the beginning to the end (Forward). Again, this is related to the visual aspect of the Show Manager, where the playback has a natural time forward direction.
  On the other hand, Sequences created with the Function Manager can have the same order properties of Chasers.
- **Editing**: The editing approach between Sequences and Chaser is different too.
  Normally the workflow of a Chaser is: create a Function, then add it to the Chaser as a step.
  The workflow of a Sequence is: create a Scene, create a Sequence on top of it, add steps to it.
  The Sequence approach might be very effective when you design a light show if you can pre-determine which fixtures you're going to use. Another major advantage of Sequence

editing is that when you create a new step, the values of the previous step are copied in the new one. So the user is simply required to adjust the differences between them.
If you are going to create 500 steps and they're all different, then Sequences and Chasers will take almost the same time to be created.

- **Synchronization**: another major advantage of using Sequences in a Show is that a Show can easily be extended (or reduced) while with a Chaser you will have a hard time to synchronize the new functions to the existing ones.
  An example. Let's say your project controls 50 fixtures that are a mixture of moving heads, PARs and scanners. At some point you buy a couple of lasers and you want them to become active in existing scenes at precise moments in time. The Show Manager allows you to do that in a few minutes ! You just need to add the 2 new fixtures to the project, add a track to the Shows affected by the change and create a few Sequences to control the lasers.
  The Show Manager resume functionality will also save you a lot of time when testing the new changes.
  With Chasers you'd probably have to deal with complex Collections and review the timings of a few steps before finding the right combination.

Show Manager toolbar controls

	Create a new Show. A Show is represented as a multitrack workspace where tracks, sequences and audio elements can be added
Shows list	This drop down box lists the currently created Shows. Clicking on a Show will display it.
	Create a new track or add existing functions to the Show. When clicking on this button a window is displayed, allowing you to perform the following operations: • Create a new track: This creates a new empty track that will serve as a container for Sequences and Audio functions. • Import an existing Scene. This will create a new track and a 10 seconds Sequence with one single step representing the selected Scene. • Import an existing Sequence: this operation scans the existing tracks and if it finds a track already bound to the Sequence's bound Scene, then add the Sequence to that track at the cursor position. If no compatible track is found, a new track will be created and bound to the Sequence's bound Scene. • Import an existing Chaser: add a Chaser function at the cursor position on the selected track. If no track is available, it will create a new one. • Import an existing Audio: add an Audio function at the cursor position on the selected track. If no track is available, it will create a new one. • Import an existing EFX: add an EFX function at the cursor position on the selected track. If no track is available, it will create a new one. • Import an existing RGB Matrix: add a RGB Matrix function at the cursor position on the selected track. If no track is available, it will create a new one. A Show can have a virtually infinite number of tracks.

	Create a new Sequence item and bind it to the selected track.
	Create a new Audio item. An audio item simply represents an audio file. Audio items can be added on any track, but if you want to to create a separate track for it, just create an new track. **Note:** It is possible to display the waveform preview of an audio item just by right clicking it and selecting the channels you want to display (mono, left channel, right channel, stereo) **Warning:** Even though QLC+ allows you to, it is not possible to play two audio files simultaneously. Especially on Windows, you might experience unwanted crashes.
	Copy the currently selected item into QLC+ clipboard
	Paste QLC+ clipboard content at the cursor position. When performing this operation two checks are performed: • Overlapping: checks if the item you're going to paste overlaps with an existing item in the selected track • Validity: If you're pasting a Sequence, QLC+ will verify that the Sequence contents are compatible with the currently selected track
	Delete the currently selected item. This can be a sequence, an audio item or a track. Note that deleting a track will delete also all its sequences/audio children. **Note: Show Manager will only perform a "visual removal" of functions. To permanently delete them, please use the Function Manager**
	Assign a custom color to the selected item
	Lock or unlock then selected item. Once an item is locked, it cannot be dragged on the timeline anymore
	Open a window where you can adjust the selected item start time and total duration. The behaviour of the latter will depend on the selected item. On Audio and Video items it does nothing. On Sequences and Chasers it will stretch all the steps timings to fit to the desired duration. On EFXs and RGB Matrices some extra options will be displayed, allowing to stretch the original function or loop the function until the desired duration is reached.
	Enable/disable the "Snap to grid" functionality. The view will be filled with vertical bars corresponding to the header markers (time or BPM) Snapping to grid will correct your items by dragging them to the nearest reference bar
00:00:00.000	This field displays the cursor time position both if the playback is stopped or activated

	Start the current Show playback from the cursor position
	Stop the current Show playback. Clicking once will stop the cursor at the current position for resuming later. Clicking twice will restore the cursor to 0 seconds. **Hint:** When resuming a show with audio tracks, please keep in mind that audio accuracy depends on the file formats you are using. For example MP3 files resume is not particularly accurate, while Wave files resume is.
Time markers	This drop down menu lets you to choose the desired time division to display for your Show. This can be either 'Time'. 'BPM 4/4', 'BPM 3/4' or 'BPM 2/2'
BPMs	When a BPM time division mode is selected, this field lets you decide the appropriate BPM to set for you Show. This ranges from 20 to 240. This can be quite useful when dealing with electronic music or BPM synchronized shows

Just 4 steps

The Show Manager has been set up to be used quickly and easily. Basically, with just 4 steps a complete Show can be created:

1. Add a new Show

First of all you need to add a new Show to the view. This creates an empty multitrack view with no tracks and no items, ready to be filled.
A popup will appear asking the name to assign to the Show. It is possible to change the name afterwards with the Function Manager, using the Show Editor panel.

2. Add a track

When you add a track, a popup will ask you to select an existing function or to create a new one with a default name.
When done, a new track will be created. All the sequences created on this track will act only on the associated Scene, not affecting any of the other tracks.
A newly created track will be set automatically as active. An active track has a green light on the left hand side.
A Track can be set to M mute and S solo states. The mute state will exclude the track from the playback, while the solo state will mute all the other tracks of the Show.
When right clicking on a track, it is possible to move it up or down for logical ordering.
Once selected, a track will display its Scene Editor on the bottom of the screen.

3. Add some Functions

When a track has been activated, you can quickly add a Sequence or an Audio

function to it by pressing the toolbar buttons.
Otherwise, by clicking on the button, it is possible to import existing QLC+ Functions and add them to the Show timeline.
A new item will be placed at the cursor position, but it is always possible to move it to the desired time just by dragging it along the timeline.
An item cannot be moved to another track, since it is bound to the track where it has originally been created.
An item can be copied , pasted or deleted with the toolbar icons. The pasted item will be placed at the current cursor position.
The item background color can be changed with the toolbar icon . The assigned color will be saved in your project file.
Once selected, an item will display its specific Function Editor on the right hand side of the screen. When right clicking an item a contextual menu will be displayed, showing the following options:

- **Preview**: This option is available only on audio items. It will display the waveform preview for right, left or stereo channels when available
- **Align to cursor**: This option will move the selected item to the cursor position
- **Lock/Unlock**: Once an item is locked, it cannot be dragged on the timeline anymore

Follow step 4 to understand how to fill a Sequence

4. Edit your Functions

Once an item has been created, it is now the time to edit it.
Please note that the difference between a Chaser and a Sequence is that a Sequence is bound to the Track (so the Scene) where it has been created. So, when adding a step, QLC+ will not ask to select a particular function, but will always use the channels of the same Scene.
When editing a Function, you will see the item changing on the multitrack view, giving an immediate feedback for aligning to other items or particular points in time.
Fade In and Fade Out times will be displayed as diagonal lines over the items on the multitrack view, while different steps/loops will be divided by vertical lines.

To increase the complexity of the Show, more Functions can be added. Just repeat the above steps depending on your needs.

And finally...play !

When a complete show has finally been created, it can be played just by clicking on the Play icon.
Playback always starts from the current cursor postion. The cursor position can be changed by clicking on the time line.

Virtual Console

The purpose of the Virtual Console is to act as a blank canvas on which the user can create a lighting desk layout of his choice. Users can place various GUI (Graphical User Interface) elements, called widgets, onto the console surface:

- Buttons for starting and stopping functions
- Sliders for adjusting channel values or function intensity
- Speed dials for adjusting function speeds
- XY Pads for manually moving intelligent light beams
- Cue lists for theatrical performance purposes
- Frames for grouping various widgets together
- Solo Frames for keeping only one button/function active at a time
- Labels to act as static information banners for other widgets

The topmost part of the Virtual Console is dedicated to a tool bar that provides quick access to the most common actions: adding new widgets, copying them to/from the clipboard, configuring the widgets and, in case of emergencies, a panic button for stopping all currently running functions.

Each Virtual Console widget's style can be configured (to some extent). Widgets can be moved anywhere and resized to any size on the Virtual Console canvas. See the article on Virtual Console Widget Styling & Placement.

Tool Bar Controls

	Add a new Button to the currently selected frame.
	Add a new Button Matrix to the currently selected frame, using the Add Button Matrix dialog.
	Add a new Slider to the currently selected frame.
	Add a new Slider Matrix to the currently selected frame.
	Add a new Animation to the currently selected frame.
	Add a new Knob to the currently selected frame. This is a convenient shortcut for adding a slider widget using the knob style.
	Add a new Speed Dial to the currently selected frame.
	Add a new XY Pad to the currently selected frame.
	Add a new Cue List to the currently selected frame.

	Add a new Frame to the currently selected frame.
	Add a new Solo Frame to the currently selected frame.
	Add a new Label to the currently selected frame.
	Add a new Audio triggers widget to the currently selected frame.
	Add a new Clock widget to the currently selected frame. This widget can be used for 3 purposes: System clock, Stopwatch and Countdown. In the last two cases and when QLC+ is in operate mode, clicking with the mouse left button will pause the counter, while clicking with the mouse right button will reset the counter to the initial value. When in system clock mode, it is possible to schedule when to start a QLC+ function during the time of day.
	Cut the currently selected widget(s) to the clipboard. Note that the widget(s) will only disappear when pasted into the new location.
	Copy the currently selected widget(s) to the clipboard.
	Paste the widget(s) in clipboard to the currently selected frame.
	Destroy the currently selected widget(s) completely. If a frame has been selected, this destroys everything inside it, including other frames.
	Configure the currently selected widget using its own configuration dialog.
	Rename the currently selected widget(s).
	Bring the selected widget to front.
	Send the selected widget to back.
	Change the background color of the selected widget.
	Set a background picture for the selected widget.
	Change the font color of the selected widget.
	Set the font properties of the selected widget.

	Configure the Virtual Console properties such as the size workspace, the widgets' default properties and the Grand Master slider properties.
	Stop all currently running functions.

Virtual Console Frame

A Frame is a container that can hold a variety of widgets, including other frames, inside. In fact, the whole Virtual Console is already your bottom-most frame. Also, if you apply some styling attributes to a frame, all of its children (which have not had their own non-default style settings applied) also inherit their parent's style properties.

Configuration

Frames can be configured with the properties button found in the toolbar or by double clicking the frame itself.

Other than the standard styling & placement options, Frames have the following additional options, divided into 2 tabs:

Appearance tab

- **Frame name**: Allow you to assign an arbitrary label to the Frame. This will be displayed only when the "Show header" option is enabled (see below).
- **Accept child widgets**: Allow you to add widgets to the Frame.
- **Allow resizing**: Allow the Frame height and width to be changed.
- **Show header**: Display the frame header. By default it will display 3 elements: An expand/collapse button, the frame name and a enable/disable button. Some further optional controls are displayed when the Frame is in multipage mode. See next paragraph.

Pages tab

Starting from version 4.5.0, QLC+ gives the possibility of turning Virtual Console Frames into multi-page widgets, useful when dealing with a lot of widgets or controllers supporting pages. The multi-page functionality is disabled by default.
Important note: this functionality will work only with an input profile with a previous page and a next page button set.
Following these options to configure this functionality:

- **Enable**: This check box enables/disables the multi-page functionality. When enabled, VC frames will display additional controls in the header bar (if the header bar is also enabled). These are: Previous page button, page number label, next page button.
- **Number of pages**: With this counter it is possible to determine the number of pages the frame is going to handle.
- **Clone first page widgets**: When checked, QLC+ will clone the widgets of the frame's first page into all the other new pages defined by the "Number of pages" field. This is a very useful option to speed up the mapping of an external controller where all the pages have the same widgets.
- **External input - Previous page**: You can set an external input signal or a keyboard combination here which will cause the previous page of the frame to be displayed.
- **External input - Next page**: You can set an external input signal or a keyboard combination here which will cause the previous page of the frame to be displayed.

Header Controls

By default a Frame is displayed with a header. Following a brief description of the default and optional controls that can be used either in Operate or Design mode.

- **Expand/Collapse button**: when clicked, the Frame size will be dramaticaly reduced, to save space on your Virtual Console. The collapsed state is saved in your project.
- **Frame name**: a label displaying the Frame name
- **Enable/Disable button**: when clicked, the Frame will go into a disable state. All the widgets inside the Frame will not respond anymore to input controls.
  This is useful, for example, to use the same input controls on widgets of different frames. For example if you bound the key "G" to a button in Frame A and also to a button on Frame B, you can enable one Frame at a time to use your key binding univocally.
- Page **Multipage controls** (Optional): when a Frame is configured in multipage mode, some extra buttons will be displayed on the header, allowing you to switch between the Frame pages and to keep track of the current page you're in.

Virtual Console Solo Frame

A Solo Frame is almost exactly the same kind of a container as a normal Frame in that can hold various widgets and other frames inside. However, the difference with Solo Frame is that it treats any Buttons inside it differently by allowing only one button to be enabled at a time. For example, consider you have **Button A** and **Button B** inside a Solo Frame with **Button A** currently enabled. Next, you click button **Button B** which automatically results in **Button A** being released, leaving now only **Button B** enabled.

Configuration

Solo Frames can be configured with the properties button found in the toolbar or by double clicking the solo frame itself.

Other than the standard styling & placement options, Solo Frames have the following additional options:

- **Frame name**: Allow you to assign an arbitrary label to the Solo Frame. This will be displayed only when the "Show header" option is enabled (see below).
- **Accept child widgets**: Allows you to add widgets to the Solo Frame.
- **Allow resizing**: Allow the Solo Frame height and width to be changed.
- **Show header**: Display a useful header composed of a button and a label. The button allows you to expand/collapse the Solo Frame, which can save a lot of Virtual Console space. The label will display the name of the Frame.

Virtual Console Button

A Button is the simplest and at the same time the most powerful widget in QLC+; with it, you can start, stop and flash your functions.

Configuration

Buttons can be configured with the properties button found in the toolbar or by double clicking the button itself.

Button label	Set the friendly name of the button. The name appears on the button in Virtual Console. **NOTE:** If you set an icon to a button with the **Edit -> Icon -> Choose** menu, the icon will override this label.
Function	Shows you the name of the Function that is currently assigned to the button. Attach a function to the button Detach the current function from the button
Attributes	Shows you the list of attributes of the assigned function that can be adjusted. Attributes can be controlled by right clicking on the button when in Operate Mode. A popup will display a number of sliders corresponding to the number of available attributes. If the assigned function is a Show, attributes correspond to the Show tracks, so you can adjust the intensity (or volume) of a whole track with just one click.
External input	You can attach an external input channel from an input device (like a slider board) to buttons so that you don't always have to use the mouse, touch screen or keyboard to access the buttons. • Input universe: The input universe that you wish to provide input data to the button. • Input channel: The individual input channel within the selected input universe that you wish to use for controlling the button. • Auto Detect: When toggled, you can just press a button on your external input hardware and it will be automatically assigned to the button. The latest combination is shown in the text boxes when QLC+ receives input data. If you don't see anything in the boxes, your input connection might have a problem that you need to fix first. • Choose...: Shows the Select Input Channel dialog that you can use to select an input channel manually.
Key combination	You can attach a keyboard key (or key combination) to the button, which then acts as if you clicked the button directly with your mouse. Attach a keyboard key (or key combination) to the button

	Detach the current key combination from the button
On button press...	• Toggle function on/off: When you click the button, the attached function is started. When you click the button a second time, the function is stopped, unless it has already stopped by itself. • Flash function: You can "flash" the attached Scene when you keep the button pressed. If another type of function is attached to the button, nothing happens when you click it. • Blackout on/off: When you click the button, QLC+ will toggle the blackout mode • Stop all functions: When you click the button, all the functions that are running in QLC+ will be stopped immediately
Adjust function intensity	If checked, this feature will adjust the intensity of the assigned function just before playing it when the button is pressed.

Virtual Console Button Matrix

A Button Matrix is basically just a way for quickly creating multiple Buttons at the same time inside a common container Frame.

When creating a new button matrix, you can assign functions to each of the buttons quite quickly with the Add Button Matrix dialog.

Configuration

Refer to Frame and Button on how to configure them.

Virtual Console Slider

Sliders are used for two distinct purposes: Fixture channel level setting and Function playback and intensity adjustment. Any slider can operate on either of these modes and each mode has its own configuration options.

Configuration

Sliders can be configured with the properties button found in the toolbar or by double clicking the slider itself.

Configuration - General Tab

The General tab holds all of the slider's properties that are shared between the slider's two modes.

Slider name	Change the slider's name.
Widget appearance	A slider can be displayed as a vertical fader or as a knob. The appearance can be changed on the fly.
Value display style	• Actual: Display actual DMX (0-255) values • Percentage: Display percentage values (0-100%)
Slider movement	• Normal: Values increase towards the top and decrease towards the bottom of the slider. • Inverted: Flip the slider upside down so that values increase towards the bottom and decrease towards the top.
External input	You can attach an external input channel from an input device (like a slider board) to sliders so that you don't always have to use the mouse or touch screen to move sliders. • Input universe: The input universe from which you wish to get input data for the slider. • Input channel: The individual input channel within the selected input universe that you wish to use for controlling the slider. • Auto Detect: When toggled, you can just move/press a button/slider/knob on your external input hardware and it will be automatically assigned to the slider. The latest combination is shown on the text boxes when QLC+ receives input data. If you don't see anything in the boxes, your input connection might have a problem that you need to fix first. • Choose...: Shows the Select Input Channel dialog that you can use to select an input channel manually.

Configuration - Level Tab

If the slider is not currently in Level mode, all you see is a button telling you to click it to switch the slider to Level mode. After you click it, the Level mode properties will be shown.

Value Range	• Low limit: Set the lowest DMX value that can be set by the slider. • High limit: Set the highest DMX value that can be set by the slider.t • From capability: You can limit the slider's level value range to a certain capability range within a fixture's channel. When you have any channel value range selected from the fixture list (for example *"Dimmer Control 006 - 128"*) and you click this button, the **low and high limit** of this slider are taken automatically from that capability and are set to 6 and 128, respectively.
Fixture list	You can select individual channels from this list that contains all channels from all of your fixtures. Those channels that you have placed the tick mark beside will be controlled by this slider.
All	Clicking this button will select ALL channels from ALL fixtures.
None	Clicking this button will clear ALL channel selections from ALL fixtures.
Invert	Invert the current selection. If you have channels 1, 3 and 5 selected from all fixtures, clicking this button will de-select channels 1, 3 and 5 and instead select channels 2, 4 and 6 from those fixtures.
By group...	If you wish to control a specific channel group from ALL fixtures, you can click this button and select the channel group to control. Accepting the dialog will select ALL channels that belong to the selected channel group from ALL fixtures. This is particularly useful if you wish to have common control over, for example, all of your scanners' *intensity* channels.
Monitor the selected channels and update the slider level	**(EXPERIMENTAL)** By enabled this option, the slider will watch the DMX channels changes and if they assume all the same level, the slider will update accordingly for an immediate visual feedback
Click & Go	Click & Go is a QLC+ technology which allows you to quickly access fixtures functionalities in a completely visual way. When Click & Go is enabled, a button will appear at the bottom of the slider and with just 2 clicks you will reach the desired result. The supported modes are: • None: Click & Go disabled. No button will be displayed. • Color: Single color selection. A gradient of the color controlled by the slider will be displayed, allowing you to jump to the desired color visually. • RGB: RGB color selection. A RGB Color picker will be displayed, allowing

Click & Go	you to pick any color from black to white. 16 preset colors are displayed on the left side for convenience. When selecting a color, the slider value will be placed half way (128). Moving it downward will fade towards black, while moving it upward will fade towards white. • Gobo/Effect: Gobo/Macro selection. A grid view of the fixture defined macros will be displayed and you will be able to choose a macro from its color, its gobo or its name. When the mouse is over a macro cell, a blue bar will appear, allowing you to choose an intermediate value within the same macro. This is useful for functionalities like Gobo rotation speed.

Configuration - Playback Tab

If the slider is not currently in Playback mode, all you see is a button telling you to click it to switch to Playback mode. After you click it, the Playback mode properties will be shown.

When the slider is in playback mode, the slider acts like a combined button and slider. You can start a function AND simultaneously control the function's intensity with the slider. When the slider is at zero, the function is stopped but any value above zero will start the function (unless it has already been started) and simultaneously adjust the function's intensity (if applicable).

Function	Displays the function that is currently attached to the slider.
	Attach a function to the slider.
	Detach the currently attached function.

Configuration - Submaster Tab

If the slider is not currently in Submaster mode, all you see is a button telling you to click it to switch to Submaster mode. After you click it, the slider will be set to work as a submaster.

When a slider is set to Submaster mode, it will control the intensity of every other widget in the same frame (please keep in mind that the main Virtual Console area is a frame too !)
The intensity of a widget depends on the type of widget itself and which functionality it controls. A submaster will control the intensity of any QLC+ "light emitting" functionality, either a function or single channel levels.
For example a submaster can control the intensity of a function attached to a button or the channel levels associated to a slider in level mode.
Every widget is controlled by a submaster even if the widget's functionality is not active yet. For example if a submaster is set to 50%, a button pressed afterward will start its associated function with 50% of intensity.

Virtual Console Slider Matrix

A Slider Matrix is basically just a way for quickly creating multiple Sliders at the same time inside a common container Frame.

Configuration

Refer to Frame and Slider for how to configure them.

Virtual Console Animation

An Animation widget is a complete tool to operate with the RGB Matrices of your project.
It displays several graphical elements to fully control a matrix during live shows.
Most importantly, it allows you to define a number of so called "custom controls" to quickly recall presets, colors and properties and apply them immediately to a running matrix.

Introduction

When clicking on the icon, an Animation widget will be added to your Virtual Console.
By default, the widget will display 4 elements:

- On the leftmost side, a slider to control the playback and the intensity of a RGB Matrix. The slider movement can be associated to an external controller input line. It behaves exactly like a Virtual Console Slider in playback mode.
- On the top right side, two Click & Go buttons to set the matrix start and end color. When clicking on them during the Operate Mode, a color picker menu will be displayed.
- Below the two buttons, a drop down menu to select the desired Matrix animation or preset.

The space below the animation menu, is reserved for Custom Controls. They can be defined in the widget configuration window and each of them will be displayed as a single button.
Each custom control can be associated to a key combination or an external controller input line.

Configuration

An Animation widget can be configured with the properties button found in the Virtual Console toolbar or by double clicking the widget itself.

General Page

Widget name	Set the friendly name of the widget, displayed above the two Click & Go buttons
Matrix Function	Shows you the name of the RGB Matrix Function that is currently assigned to the widget. Attach a RGB Matrix Function to the widget Detach the current function from the widget
Apply color and preset changes immediately	Select whether the changes to the Matrix must be applied immediately or at the next Matrix loop
	You can attach an external input channel from an input device (like a slider board) to the widget slider so that you don't always have to use the mouse or

External input	touch screen to control the Matrix playback or intensity. • Input universe: The input universe from which you wish to get input data for the slider. • Input channel: The individual input channel within the selected input universe that you wish to use for controlling the slider. • Auto Detect: When toggled, you can just move/press a button/slider/knob on your external input hardware and it will be automatically assigned to the slider. The latest combination is shown on the text boxes when QLC+ receives input data. If you don't see anything in the boxes, your input connection might have a problem that you need to fix first. • Choose...: Shows the Select Input Channel dialog that you can use to select an input channel manually.

Custom Controls Page

A custom control is a "shortcut" to a functionality of a RGB Matrix.
Basically you can control all the options that are normally used in the RGB Matrix Editor during the design pahse of a show. The only option you can't control is the Fixture group used by a RGB Matrix Function.
In this configuration page the user can define any number of custom controls depending on the needs during a live performance.
Each custom control is represented as a button in the Animation widget layout and can be either clicked with the mouse or pressed on a touchscreen, or associated to a key combination or an external controller input line.

Add start color	Add a Matrix start color control. When clicking this button, a color selection tool will be displayed. The button displayed in the widget's layout will have a background color representing the one selected and the letter **S**.
Add end color	Add a Matrix end color control. When clicking this button, a color selection tool will be displayed. The button displayed in the widget's layout will have a background color representing the one selected and the letter **E**.
Add end color reset	Add a Matrix end color reset control. The button displayed in the widget's layout will have a gray background color and when pressed, it will reset the current matrix end color (if previously set)
Add preset	Add a Matrix preset control. When clicking this button, a dialog will be displayed for the preset selection. The button displayed in the widget's layout will have a gray background color and a label corresponding to the preset's name.
Add text	Add a Matrix text animation control. When clicking this button, a dialog will be displayed requiring to enter the desired text to be animated on the Matrix. The button displayed in the widget's layout will have a gray background color and a label corresponding to the chosen text.

Remove	Remove the currently selected control from the controls list.

Virtual Console Speed Dial

Speed dials are meant for adjusting the speed of a set of functions from virtual console, much like bus-mode sliders in the previous versions of QLC+. While the bus concept relied on assigning functions and a slider to the same bus, the speed dial's method is a simpler and more straightforward one: The user directly selects a set of functions whose speed properties are controlled with the dial, thus removing the need for buses in the middle.

Operation

The speed dial itself looks and works just the same as the speed dials in various function editors: There is a **big dial** in the middle that increases the time when rotated clockwise and decreases it when rotated counter-clockwise. The plus and minus buttons are an alternative way of adjusting the speed.

The **current time** value is displayed in four parts. When any of these fields is clicked, the dial and the plus/minus buttons will increase/decrease that unit of time. The mouse wheel can also be used instead of the dial/buttons and the fields also accept values entered on the numeric keypad.

- **h**: Hours
- **m**: Minutes
- **s**: Seconds
- **.xx**: Centiseconds (100ths of a second)

Clicking the **infinite** button disables all of the dial's controls and sets the currently adjusted functions' speed to infinity. For example, with a Chaser, an infinite duration means that it will never switch to the next step unless manually instructed to (see Cue List).

The **tap** button can be used to match the time to a beat. The elapsed time between clicks is measured and this time is set for the controlled functions, just as if the dial had been adjusted to that particular time.

For cuelists/chasers and RGB Matrices, the tap button advances things to next step/position when speed dial is set to adjust duration. The tapped tempo must be more than 1/4 of the original duration.

Configuration - Functions tab

Speed dials can be configured with the properties button found in the toolbar or by double clicking the speed dial itself.

Speed Dial Name	Change the dial's name.
	Displays the list of functions, whose speed value(s) are being controlled by the dial. For each speed value (fade in, fade out, duration), it is possible to set a

Function table	multiplier. **Fade In *** — Select a multiplier to adjust the functions' fade in speed **Fade Out *** — Select a multiplier to adjust the functions' fade out speed **Duration *** — Select a multiplier to adjust the functions' duration This multiplier will be applied to the speed dial value before adjusting the functions' speed value. Select **(Not Sent)** so the functions' speed value will not be affected by this speed dial.
	Add function(s) to be controlled by the speed dial.
	Remove the selected functions from the speed dial's list of controlled functions.

Configuration - Input tab

External input can be configured here: one for value with minimum and maximum and another for tap button.

Configuration - Appearance tab

Individual visibility of the widget's layout elements can be switched on or off here. This way it is possible to save some screen real estate if some of the fields are not needed (e.g. when controlled by an external controller, the big dial is not needed).

Show the plus and minus buttons
Show the central dial
Show the tap button
Show the hours field
Show the minutes field
Show the seconds field
Show the milliseconds field
Show the inifinite option

Virtual Console Cue List

A Cue List provides a list of Functions that you can step through with a single keyboard key. The Cue List is designed for theatrical performances where the lighting operator needs only to follow the script and toggle the next cue as the performance goes on.

The first column on the Cue List displays the step (cue) number that runs from 1 to infinity and beyond . The second column shows the individual Function name that has been assigned to that particular cue.

Please note that only Chasers can be assigned to a Cue List, for the simple reason that the Chaser is the only function with a duration. Since any other Function can be added as a step to a Chaser with the Chaser Editor, it is simple to create the desired Cues by mixing Scenes, Collections, and so on...

Configuration

Cue Lists can be configured with the object properties button found in the toolbar or by double clicking on the Cue List widget.

Cue list name	Set the friendly name of the Cue List. The name appears as the Cue List's second column title.
Cue list	This panel allows the user to select the Chaser to associate with the Cue List.
	Associate a Chaser with the Cue List using the Select Function dialog.
	Remove the selected Chaser from the Cue List.
Playback	You can select the key that you want to use to start/stop the selected step (cue) in the Cue List with this option. The key combination is shown in the text field; if the field is empty, there is no key combination currently attached to the Cue List.
Next cue	You can select the key that you want to use to skip to the next step (cue) in the Cue List with this option. The key combination is shown on the text field; if the field is empty, there is no key combination currently attached to the Cue List.
Previous cue	You can select the key that you want to use to skip to the previous step (cue) in the Cue List with this option. The key combination is shown on the text field; if the field is empty, there is no key combination currently attached to the Cue List.
Crossfade	You can select external input for the two crossfaders.

Operate mode

When switching QLC+ to Operate Mode, the Cue List will become active, allowing the selection of the desired steps included in the associated Chaser.

Pressing the ENTER key will start the selected step.

The following items are displayed at the bottom of the Cue List widget:

Cue progress bar	A progress bar showing the current running step status. When a fade in speed is set and is not set to "default" the bar will be colored in green. During the hold time the bar will be colored in blue. The text displayed on the bar is the time remaining for the cue to end. If the step duration is set to inifinite (∞), the bar will just display the fade in progress in green.
	When clicked, this button will open/close a side panel showing two sliders for controlling crossfade. (See next chapter)
	Start/stop the replay of the Cue List from the currently selected step
	Go back to the previous step in the Cue List, which will then be replayed. When the top is reached, the previous step will be the last step of the Cue List
	Go to the next step in the Cue List, which will then be replayed. When reaching the bottom, the next step will be the first of the Cue List

Crossfade

There are two faders that can be used to manually control crossfade between two consecutive steps.

Each slider is connected to a Cue List step and to help to understand their purpose we'll call them "current step" and "next step" sliders.
The label at the bottom of each slider will show the number of the step that the slider is controlling and a color to show the status of that step.

The checkbox at the top will link the sliders together to facilitate crossfading between steps.
When the Cue list is running the sliders control the intensity of the associated steps, overriding their Fade In and Fade Out speeds and so allowing manual control of the transition.

The slider for the current step has a blue label. This will be at 100% when the replay of the Cue List is started.
The slider for the next step has an orange label and will be at 0% when the replay of the Cue List is started.
The intensities of both steps can be adjusted by moving the corresponding slider.

However, after both sliders have been moved to the opposite ends of their travel, the following changes take place:
The orange next step label changes to blue (slider at 100%). This indicates that the old next step has become the new current step.
The blue current step label changes to orange and the number of the step is increased by 2

(slider at 0%). So this slider now controls the step which comes after the old next step.

Hint

When a step is added to a Chaser using the Chaser Editor, the default duration is set to 0.
To avoid the steps in the Cue List looping frantically without any result, set the duration of your steps, either by double clicking on the duration field or by using the Speed Dial widget
Note that if you need scenes that you're going to crossfade manually with the Cue List widget, you probably will want to set the duration of the steps to "infinite" (∞) using the Speed Dial widget. This can be enabled in the Chaser Editor by clicking on the button.

Virtual Console XY Pad

An XY Pad is for manually controlling the **pan** and **tilt** channels of intelligent lighting fixtures, namely scanners and moving heads. The pad is a resizable canvas with a handle and two sliders that you can grab to obtain direct access to the attached fixtures' mirror/head position.

Controls

(1) Main area	This is the area where the movement occurs.
(2) Handle	The gray point is the handle. You can move it either with the mouse or with the keyboard.
(3) Range sliders	These sliders limit the working area.
(4) Value sliders	These sliders can be used to change value in one axis only.
(5) XY Pad name	The name can be changed in properties, see below.
(6) DMX Coordinates	This part displays current DMX coordinates in format (Pan course.Pan fine : Tilt course.Tilt fine).
(7) Horizontal and vertical angle	If the fixtures controlled have proper pan and tilt range filled in in their definitions, and the range is same for all fixtures, then this field will compute angle from DMX coordinates. 0° is in the middle of the range (DMX value 127.127). For example, if pan range for a fixture is 540°, the displayed angle will be between -270° and 270°. Note: this may not work properly, when a fixture has limited axis range in configuration, or reversed axis.

Range limiting

Starting from QLC+ version 4.5.0, two more sliders -- **(3)** in the picture above -- have been introduced to give you the opportunity to set a working range window in which your moving heads will operate.
Note that from the GUI the movement of the handles will be limited to the green window area, while with an external controller all the values will be scaled to the window, so you will be able to have more sensitivity in the control your fixtures.

In addition, it is possible to set limits for individual fixtures in the properties dialog, see Configuration section. This setting maps the whole range of the area (the grey area) to the sub specified range.

Example: set fixture X Axis (=pan) mininum to 20% (DMX value 51), maximum to 80% (DMX value 204). When the handle is at the left edge (value 0), actual DMX output is 51. Similarly, handle at the right edge will output 204 (=80%). For values in between, DMX output is scaled proportionally.

It is possible to have both limits enabled (using the range sliders and limits per fixtures).

Configuration

XY Pads can be configured with the properties button found in the toolbar or by double clicking on the XY pad itself.

XY Pad name	Set the name of the XY Pad. The name **(5)** is shown in the upper left hand corner of the widget on Virtual Console.
Y-Axis slider movement	The appearance of the vertical slider (the Y-Axis control) can be set to **Normal** or **Inverted**. In the first case the maximum value will be reached at the bottom of the pad, while in the second case it will be reached at the top.
Fixture tab	Shows the fixtures that are currently controlled with the XY Pad. • Fixture: The names of each fixture • X-Axis: Shows the value range (and reversal if applicable) of the horizontal (pan) axis for each fixture • Y-Axis: Shows the value range (and reversal if applicable) of the vertical

	(tilt) axis for each fixture
	Add Fixtures to be controlled by the XY Pad, using the Select Fixture dialog. **NOTE:** only those fixtures that have pan and tilt capabilities are shown in the dialog.
	Remove the selected fixtures from the XY Pad's control list.
	Edit the selected fixture's behaviour as controlled by the XY Pad.

Fixture movement configuration

You can change the extent of movement and reverse fixtures' movement with the XY Pad Fixture Configuration dialog.

Horizontal X-Axis	• Minimum: Set the minimum pan limit for the fixture • Maximum: Set the maximum pan limit for the fixture • Reverse: Reverse fixture's movement on the horizontal (pan) axis
Vertical Y-Axis	• Minimum: Set the minimum tilt limit for the fixture • Maximum: Set the maximum tilt limit for the fixture • Reverse: Reverse fixture's movement on the vertical (tilt) axis

Keyboard control

It is possible to control head position with arrow keys on the keyboard. Each keypress increases/decreases coarse value by 1. With Shift key pressed, fine channel is changed by 1. With Ctrl, the step is 10 instead of 1.

Arrow keys	Coarse	step=1
Shift + Arrow keys	Fine	step=1
Ctrl + Arrow keys	Coarse	step=10
Shift + Ctrl + Arrow keys	Fine	step=10

External input configuration (Input Tab)

You can attach an external input channel from an input device (like a tablet) to XY pads so that you don't always have to use the mouse to control your fixtures.

- Input universe: The input universe that you wish to get input data for the XY pad.
- Input channel: The individual input channel within the selected input universe that you wish to use for controlling the XY pad.
- Auto Detect: When toggled, you can just move/press a button/slider/knob on your external input hardware and it will be automatically assigned to the XY pad. The latest combination is shown on the text boxes when QLC+ receives input data. If you don't see anything in the boxes, your input connection might have a problem that you need to fix first.
  Hint: When assigning a XY pad from OSC, you need to click the "Tilt / Vertical axis" auto detect button, otherwise the X and Y axes will be swapped.
- Choose...: Shows the Select Input Channel dialog that you can use to select an input channel manually.

Virtual Console Label

A Label is simply text field that can be placed on the Virtual Console to give additional information, for example, as a heading for a group of buttons or the title of a frame.

Configuration

You can only set up the text to be displayed in a Label with the **Rename** button. It does not have any other purpose nor configurable options, other than the standard styling & placement options that all Virtual Console widgets have.

Virtual Console Audio Triggers

Starting from QLC+ version 4.4.0, this functionality allows you to use an audio input source such as a microphone, to add more life to your light shows.

Introduction

When clicking on the icon, an audio trigger widget will be added to your Virtual Console. The widget's graphics area shows live monitoring of the captured audio, displaying a number of spectrum bars and a volume bar.
On the bottom you can see the range of frequencies analyzed by QLC+

Configuration

When double clicking on the widget in design mode, a panel is displayed showing a complete set of options to tune the audio trigger's functionality.
The first thing you can configure is the number of spectrum bars that you want to display and that you will need during your live show. The number accepted is between 5 and 32.
Once the number of bars has been decided, you can proceed to assign a functionality to each bar. There is a list showing the following options for the volume and spectrum bars:

Name	Can be "Volume Bar" or a spectrum bar shown as #number (start frequency - end frequency). Example: #5 (1250Hz - 1562Hz)>
Type	Indicates the type of functionality the bar will control. It can be: • None - No functionality assigned • DMX - Controls single DMX channels • Function - Controls a QLC+ function • VC Widget - Controls a Virtual Console widget (at the moment only buttons, sliders, tapping of speed dials and next cue of cue lists)
Assign	When a type is selected, a button with a icon is displayed to allow you to connect the bar to the desired functionality. Depending on which type you chose, clicking on this button will display the DMX channels, Functions or Virtual Console widgets selection dialogs.
Info	This column displays some additional information about the bar--functionality association. Depending on the type you chose, you will find the number of DMX channels, the function name or the VC Widget name selected displayed here
Disable	When selecting a Function or a VC Widget button, this column allows you to set a

Disable threshold	deactivation threshold percentage. When the spectrum or volume bar goes below this value, the associated function/VC button will be stopped/deactivated.
Enable threshold	When selecting a Function or a VC Widget button, this column allows you to set an activation threshold percentage. When the spectrum or volume bar goes above this value, the associated function/VC button will be started/activated.
Divisor	For speed dials and cue lists, this will divide the taps by the selected number - only every n-th tap will be actually sent to the speed dial. In other words, the tap happens only on every n-th beat. If you want the tap on every beat, enter 1, for every other beat enter 2. For once in a 4/4 measure, enter 4. For once in 3 4/4 measures, enter 12. Maximum is 64.

DMX channels

You can choose one or more channels of the currently patched fixtures. Those channels will be set proportionally to total volume, or volume in a particular frequency band.

Functions

You can select one or more functions. Those functions are started when volume goes above the Enable threshold in respective band, and stopped when it goes below the Disable threshold.

VC Widgets

You can select only **ONE** widget here, either a button, a slider or a speed dial. Depending on the type:

- **Button** is pressed when volume goes above Enable threshold, and released when the volume is below Disable threshold (similar to functions).
- **Slider** is moved in proportion to the volume (similar to DMX channels). For a slider the thresholds do not apply.
- **Speed dial** is tapped.
- **Cue List** "Next Cue" is pressed.

For **Speed dials** and **Cue Lists**, the tap/button press happens when the volume goes above the Enable threshold, and won't happen again until the volume falls below the Disable threshold. This probably means you will have to set the thresholds a little closer to each other than for buttons.

Select Input Channel

Whenever an Input Channel needs to be selected, for example when manually attaching a channel from an Input Plugin to a Virtual Console widget, the Select Input Channel dialog is used.

The dialog is very straightforward; there is a list of input universes and each universe that has an Input Line attached to it, displays the line's (or the attached device's) name after the universe number. Under each operational universe, there is a list of channels provided by that universe. You can choose from these channels or select one manually.

If you have an Input Profile attached to the universe, you see the individual channel names and numbers for the device. If there isn't a Profile attached to an input universe, you will need to type the channel number manually, by double-clicking the item that tells you to do that and then type in the channel number manually. See also the tutorial on input profiles

NOTE: All Virtual Console widgets that have been made *external-input-aware*, provide an option for automatically detecting the input channel as well, so that you don't need do this selection manually.

Virtual Console Widget Styling & Placement

The colors, font, frame style, stacking order, size and placement of widgets can be changed to better suit your vision of the perfect lighting console. Most of these options, except for moving and resizing, are available in the Virtual Console **Edit** menu, that is, they can be accessed from the menu bar as well as from a popup menu by right clicking a widget you wish to adjust.

NOTE: The Virtual Console widgets' styling and placement options are available only when the application is in Design Mode.

Moving Widgets

Widgets can be moved simply by dragging them as if you were dragging any icon on your desktop. Refer to your operating system manual on how to drag items.

To move a widget from one frame to another frame, you need to move the widget through the Virtual Console clipboard. Cut the widgets from the first frame, then select another frame and paste the widgets into that frame.

Resizing Widgets

Each widget has a special resize handle on their lower right hand corner that you can grab and then resize the widget as appropriate simply by dragging the handle.

Widget Background

To change a widget's background color, choose **Color** from the **Background** menu and select a color from the popup dialog. To set an image as the widget's background, choose **Image** from the **Background** menu and select an image file from the popup dialog.

To restore the default background color (and to clear an image), choose **Default** from the **Background** menu.

NOTE: Some operating systems and/or visual themes do not allow manual setting of background colors or images. Try changing the desktop theme if you encounter this problem.

Widget Foreground

To change a widget's foreground color, choose **Color** from the **Foreground** menu and select a color from the popup dialog.

Buttons can have an icon in the foreground, which will override any text set as their name. To change a button's icon, select **Choose...** from the **Icon** menu and select an image file from the popup dialog.

To restore the default background color (and to clear an icon if applicable), choose **Default** from the **Background** menu.

NOTE: Some operating systems and/or visual themes do not allow manual setting of foreground colors or images. Try changing the desktop theme if you encounter this problem.

Widget Borders

To change the border style for a widget, you can choose either **Raised** or **Sunken** from the **Border** menu. To completely remove the borders of a widget, choose **None** from the **Border** menu.

NOTE: Some operating systems and/or visual themes do not allow manual setting of borders. Try changing the desktop theme if you encounter this problem.

Fonts

You can change the font for all child widgets of a frame at the same time by changing the font of the frame itself. Alternatively, you can change the font for each widget separately. Fonts can be changed by selecting **Font...** from the **Font** menu. To restore the font back to the widget's parent frame's font, select **Default** from the **Font** menu.

Widget Stacking Order

In addition to the normal 2-dimensions, vertical and horizontal, you can also adjust widgets' placement in the 3rd dimension: depth. If you see a widget going behind another control, you can **raise** the widget on top of all other widgets by selecting **Raise** from the **Stacking Order** menu. Likewise, if you wish to put a widget behind all other widgets, you can select **Lower** from the **Stacking Order** menu.

Simple Desk

The simple desk emulates a typical lighting console that is able to control a full 512-channel DMX universe with a multiple cue stacks operated with playback sliders.
Simple desk is divided into two main areas: **channels** (top area) and **cues** (bottom area).
The channels area reflects exactly the current status of each DMX channel of each universe. If channels are controlled with no project loaded, users have a completely manual control of them. When a project is loaded, instead, moving a Simple desk channel will override any other QLC+ function trying to control that channel. A visual indication will turn the channel background to red, informing the user that Simple desk now has the total control of it.
This is very useful in some live circumstances where a running function needs some manual adjustment.
To "disengage" Simple desk from overriding channels, just click on the reset button. Resetting channels will either bring them back to zero or to the value previously set by a function.

Cues operates separately from other QLC+ components. For example, cues within the cue stack are not visible in the Function Manager and Scenes are not visible in the cue stack.

Controls - Universe

The universe box contains the sliders that are used to control individual DMX channels in the first DMX universe. Since 512 sliders cannot fit nicely on the screen at the same time, they have been divided into pages. By default, each page contains 32 sliders but that can be tuned.

	Switch the view mode from all channels to fixture channels. Please note that if no fixtures has been defined, the second mode will produce an empty result
	Skip to the previous DMX page.
Universe page box	Displays the current DMX page. You can skip to a page by writing the page number directly into this box or use the mouse wheel to skip between pages quickly.
	Skip to the next DMX page.
	Reset all DMX sliders back to zero or to the value previously set by a function.
GM	The Grand Master

Controls - Playback

The playback box contains a bunch of playback sliders each of which may contain a cue stack. The playbacks can be used to "play back" the contents of their cue stacks and they can also be used to adjust their overall intensity of the cue.

	Select the currently-active playback, whose cue stack is shown on the right side of the screen.
Playback slider	Adjust the intensity of playback's cue stack. When the slider is drawn all the way to zero, the cue stack is stopped. Any value above zero will start replaying the playbacks' cue stack.
	Flash the playback's first cue.

Controls - Cue Stack

The cue stack box shows the contents of the currently selected playback.

	Skip to the previous cue (or start playback on the last cue in the cue stack with full intensity).
	Stop the currently active cue stack.
	Skip to the next cue (or start playback on the first cue in the cue stack with full intensity).
	Switch to/from cue edit mode. When this button is pressed, you can edit the contents of individual cues; the currently active cue's contents are shown on the DMX sliders. The Fade In speed, Fade Out speed and Duration as well as the name of individual cues can be adjusted with the speed dials that are displayed in a separate tool window. You can also select multiple cues to adjust their speeds all at the same time, but then the DMX sliders are disabled to prevent you from accidentally overwriting all cue contents with the same values.
	Record a new cue, taking its contents from the current DMX slider values.
Cue Stack	This box displays the contents of the currently selected playback's cue stack, along with the index number (1, 2, 3...), Fade In speed, Fade Out speed, Duration and an optional name for each cue. You can change the cue order by dragging the cues on top of each other, either one at a time or multiple cues at the same time.

Tuning

Please refer to the Manual parameters tuning Simple Desk section

Input/Output Mapping Howto

This howto document tells you how to patch plugins and their input/output lines (physical input/output devices) to QLC+'s universes.

By default QLC+ provides 4 universes but you can add/remove them as needed.
The input/output mapping is saved in the currently loaded project. This allows you to port your project on another computer/OS without the need to reconfigure it every time.
If no project is loaded, QLC+ will keep the I/O mapping as a "fallback" configuration.

Input/Output Manager

To access the Input/Output Manager, just click on the tab with the icon placed on the bottom of the QLC+ main screen.
The screen is composed in this way:

- On the left hand side there is the list of internal universes that QLC+ can manage
- On the right hand side there is the list of devices and their mapped inputs, outputs and feedback lines that QLC+ has detected
- On the bottom right hand side there is a panel displaying brief information on the currently selected device

Every device has a checkbox whenever an input, output or feedback line is available.
Each QLC+ universe can map a single input, a single output and a single feedback line

Some plugins might require configuration before they can be used' so you might not be able to see all inputs/outputs at first. The configuration button is place next to the information panel and it is enabled if the plugin allows any manual setting.

The button is icon is:

Adding/Removing universes

QLC+ supports any number of universes, depending on the CPU limit of the device controlling them.
On the left hand side of the Input/Output Manager there is a toolbar where you can add/remove, name and configure universes.

	Add a new universe. The universe will have a name like "Universe X", where X is a progressive number assigned by QLC+ (and also the Universe ID).
	Remove the currently selected universe. **Please be careful with this operation as it can compromise your project and cannot be reverted.** When deleting a universe, if it is currently patched or some fixtures are mapped on it, a popup message will appear asking for confirmation if the operation should be completed or abandoned.

Universe name	An arbitrary string that you can set to quickly identify the meaning of a Universe
Passthrough	Set the universe to just forward what it receives in its input line to its output line This is useful when you want to use QLC+ to act as a "protocol" converter. For example you can use this feature to transparently map an ArtNet network to a DMX USB adapter.

Patching

To patch a plugin's input/output line to the selected universe, you need to place a checkmark on that particular plugin's input/output line. You can have only one line assigned to a universe at a time, so when you check another line, the checkmark will move from its previous position to the one you just checked.
If you don't see any line on a plugin, it means you don't have any device that QLC+ understands and you're left with the one and only (non-selectable) choice: None.

When an input/output line is checked, the corresponding universe information on the left hand side of the screen will change and will display the new configuration set.
The plugin information on the bottom right hand side of the screen will change as well and will give you the new status of the plugin line.

Input and Feedbacks

When a plugin input line is checked, it gets enabled right away, so you can perform a basic test to double check if your hardware is working properly with QLC+.

Just move a fader/knob on your external device, and if everything works fine, you will see a icon appearing beside the corresponding universe on the left side of the screen.

If your input device supports a return channel, QLC+ can send a visual/mechanical feedback to it. Devices such as Behringer BCF2000 support this feature.
At the moment feedbacks are supported only through MIDI.

To learn how to setup your external input device for the best use with QLC+, please continue your reading with the howto for input profiles.

Input Profiles Howto

This howto document tells you how to associate input profiles to input universes and how to edit these profiles. You should first read the howto on input/output mapping so that you can access the input/output manager and know how to edit input universes.

Input Profile Manager

To access the Input/Output Manager, just click on the tab with the icon placed on the bottom of the QLC+ main screen.
Select the desired universe on the left part of the screen and then click the Profile tab placed beside the Mapping tab on the top right part of the screen.

Profile assignment

You will see a list of available input profile definitions, with a check mark on None. This means that the current universe doesn't have a profile assigned yet. To assign a profile to the universe, simply place a check mark on one of the available profiles by clicking on the empty check box. Only one profile can be assigned to one universe at a time, so the check mark will actually move from its previous position to the item that you have just selected.

KORG nanoPAD: For unknown reasons, the nanoPAD factory defaults don't map the X Axis of the pad area. To have it fully working with QLC+, please download the KORG utility (Windows and OSX only) from here and set the X Axis to CC2 (Control Change #2).

Add/Edit a Profile

It is probable that your input profile is not on the list and you need to create one of your own.

Click the create new input profile button to start making a profile definition for your input profile. Alternatively, you can edit any existing profile by selecting the appropriate item and clicking the edit button. The procedure is exactly the same in both cases from now on.

Input Profile Editor

A dialog is opened with entry fields for Manufacturer, Model and Type.

First enter the profile's manufacturer and the model to these fields and select profile type.

Type is one of:

- MIDI - for MIDI profiles, usually used with MIDI plugin
- OSC - for OSC profiles, used with OSC plugin
- HID - for HID profiles, used with HID plugin
- DMX - generic DMX profiles

- ENTTEC - ENTTEC Wing profiles, used with Enttec Wing plugin

So far, the only difference is that MIDI profiles show MIDI message parameters in the channel editor.

Now click the Channels tab to edit the profile's channels. You have two choices for adding channel definitions: Manual and Automatic:

Manual mode

Click the add button to enter individual channel information by hand for each channel.

Click to remove an existing channel

Automatic mode

Click the automatic wizard button to attempt automatic channel detection. You'll receive further instructions from QLC+. You must have an input plugin assigned to the current universe for this feature to work. Also, you must first stop the wizard to be able to navigate away from this dialog page.

Channel properties

When you add or edit a channel, a small window will be displayed, asking you to fill or change some parameters:

- **Number**: The channel number. Since QLC+ supports a wide variety of input plugins, the channel number might not be intuitive, so only edit this if you know what you're doing.
- **Name**: The channel name. This is an arbitrary string to indicate the purpose of a channel.
- **Type**: The channel type. This can be: Slider, Knob, Button

  Other types: Previous page, Next Page, Set Page were used for Multipage frames.

For MIDI profiles the dialog contains additional fields:

- Channel
- Message
- Param
- Note

where you can enter the channel specification (which translates to channel number) in a more intuitive way.

Note that you cannot add the same channel multiple times to one profile.

Sliders movement properties

If your input profile includes slider channels , when you click on them you'll notice some extra properties showing up at the bottom of the input profile editor main window. With those, you can set how values received from a slider should act within QLC+.

There are two behaviours: Absolute and Relative.

Absolute is the default setting and basically tells QLC+ to use the slider values exactly as they are received from an external controller.
Relative is a more advanced behaviour that comes handy when using a HID Joystick with a QLC+ XY Pad widget or a Slider widget. Values received from an external controller are treated as relative movement starting from the current position of a Virtual Console widget.
Let's make an example. Suppose you have a XY Pad in your Virtual Console, controlling and monitoring a group of moving heads. During your show you will have a number of scenes moving the heads pan and tilt. At some point you want to slightly adjust the position of the heads of just a few degrees. Here is when the relative movement kicks in. When you move your joystick (or external slider) QLC+ will adjust the heads from their current positions. The direction will depend directly from your external controller. The relative movement will stop when the external controller will return to its origin. Joysticks have a spring for that.
In addition to this, the Input Profile Editor Relative setting allows you to set a **Sensitivity** parameter that will instruct QLC+ about the strenght of your external controller movements. The higher this value is, the slower the movements will occur. The lower, the faster.

Back to the input profile definitions panel

When you're done with channels mapping, click the OK button to accept changes and save the input profile. If you didn't enter a manufacturer/model for the profile, you'll be prompted to enter them before you can continue.
Now you should see the profile you just defined in the list of available input profiles. Remember how to assign it to the current universe? Scroll up to Profile assignment if you don't.

To remove any existing input profiles, click the remove button. Note that some profiles are so-called system profiles and cannot be removed unless you're the administrator.

That's all !
Now you can start using your preferred profile. When assigining an input channel to a QLC+ element (like Virtual Console sliders, channel groups, etc..) you will see that your Input profile mapping will be used.

List of bundled input profiles

Device	Type	Comment
Akai APC20	MIDI	use APC20 miditemplate
Akai APC40	MIDI	use APC40 miditemplate
Akai APC Mini	MIDI	no miditemplate needed

Behringer BCF2000 in default mode	MIDI	
Behringer BCF2000 in Mackie Control mode	MIDI	has more controls than the default mode
Behringer LC2412	MIDI	
Elation MIDIcon	MIDI	not tested
Enttec Playback Wing	Enttec	
Enttec Shortcut Wing	Enttec	
Generic MIDI	MIDI	
Korg nanoKONTROL	MIDI	
Korg nanoKONTROL 2	MIDI	
Korg nanoPAD	MIDI	
Logitech Wingman Attack 2	HID	
Novation Launchpad	MIDI	
Showtec Showmaster 24	MIDI	should work with lot of similar devices: Elation Light Operator 24, Stairville LC24, and most probably also with their 48 channel variants
TouchOSC Mix16	OSC	

ArtNet input/output plugin

Introduction

QLC+ supports the ArtNet protocol through an input/output plugin that receives and transmits packets on the network.
No extra requirements are needed, since QLC+ has a native implementation of the ArtNet protocol that works on Linux, Windows and OSX systems.
The ArtNet plugin can send and receive packets from multiple network cards, virtual addresses and multiple universes per network card.
When transmitting multiple universes on the same interface, the packets will be sent with an ArtNet Universe ID equal to the QLC+ universe number minus 1.

For example:
QLC+ Universe 1 --> ArtNet Universe 0
QLC+ Universe 2 --> ArtNet Universe 1
...
QLC+ Universe 8 --> ArtNet Universe 7

This choice is due to some facts:
1- The first valid ArtNet universe is 0 and not 1
2- The first universe accepted by commercial ArtNet-DMX devices like eDMX and ODE is 0, so to have QLC+ to work out of the box, the first ArtNet universe must be 0.

Configuration

The ArtNet plugin doesn't really have a configuration panel. When clicking on the configuration button, a panel will be shown, displaying the ArtNet nodes discovered, grouped by network interface.

DMXKing eDMX and ENTTEC ODE

If you own one of these devices, they both have configuration tools that might come handy when working with QLC+. With them you can set several parameters to match the best configuration to input/output data from/to QLC+.
For example if you want QLC+ universe 3 to output data on the first port of a eDMX, you need to use the tool below to change the device universe address to 2.

Here the links to download the tools:
DMXKing eDMX Configuration tool
ENTTEC Node Management Utility

Compatibility

QLC+ has been tested with the following ArtNet softwares/devices:

- DMXking eDMX1 TX - 1 Output device
- DMXking eDMX2 TX - 2 Outputs device
- Enttec Open DMX Ethernet (ODE) - Input/Output device
- OLA - Open Lighting Architecture - Input/Output software node
- ArtNet Controller LITE - Input controller for Android
- Controlador ArtNet DMX (Lite) - Input controller for Android
- Modul8
- Jinx! - LED Matrix Control

DMX USB input/output plugin

1 Introduction

The DMX USB plugin supports a variety of FTDI-based USB-to-DMX devices:

- DMXKing USB DMX512-A
- DMXKing ultraDMX micro
- DMXKing ultraDMX Pro
- Enttec DMXUSB Open
- Enttec DMXUSB Pro
- Enttec DMX USB PRO Mk2
- DMX4ALL USB-DMX STAGE-PROFI MK2
- DMX4ALL NanoDMX
- Gus Electronics USB-VL344
- Vince DMX512 USB
- ElectroTAS USB-DMX
- FTDI USBCOM 485+

2 Configuration

DMX USB devices should be automatically detected from QLC+ and displayed in the input/output panels list.
If for some reason the auto-detection fails, you can "force" the type of your DMX USB adapter manually.

Click on the name of your device and open the configuration dialog by clicking on the icon on the bottom-right side of the panel.
You will see a list of DMX USB devices currently connected to your computer. Each one has a drop down menu where you can force the device type.
Here's the meaning of each one:

- **Open TX**: Enttec USB DMX Open (and clones) in output mode
- **Pro RX/TX**: Enttec USB DMX Pro or most of the DMXKing devices
- **Pro Mk2**: Enttec USB DMX Pro Mk2 - 2 DMX outputs, 1 DMX input, 1 MIDI IN and 1 MIDI OUT ports are available
- **Ultra Pro**: DMXKing ultraDMX Pro with 2 outputs and 1 input
- **DMX4ALL**: DMX4ALL USB-DMX STAGE-PROFI MK2
- **Vince Tx**: Vince DMX512 USB in output mode

3 Requirements

2.1 Linux

On all Linux distributions, you need to install libftdi. If you install QLC+ with the Ubuntu Software Center or some other automatic installer tool, this library will be installed automatically for you.

In some cases, if the device doesn't output anything, it might be useful to add your user to the "dialout" group with the following command:

```
sudo adduser your_user_name dialout
```

2.2 Mac OS X

On the Apple Mac OS X, you don't need any drivers at all since QLC+ uses the OS X native USB interface. Installing the D2XX drivers should cause no harm, but **DO NOT INSTALL VCP drivers** as they will definitely interfere with QLC+. If you have previously installed the VCP drivers, consult the FTDI installation guides on how to uninstall them.

OSX Mavericks issues: The problem is in a new Apple USB driver. The procedure to disable it follows below.
Please note that this can compromise the behaviour of other USB devices, so do it only if you know what you're doing!
From a terminal type the following commands:

```
cd /System/Library/Extensions/IOUSBFamily.kext/Contents/PlugIns
sudo mv AppleUSBFTDI.kext AppleUSBFTDI.disabled
sudo touch /System/Library/Extensions
```

2.3 Windows

On Microsoft Windows, the plugin needs the latest D2XX drivers from FTDI. Consult the FTDI installation guides on how to install the drivers. **DO NOT INSTALL VCP drivers** as they will probably interfere with the D2XX interface.

4 ENTTEC DMX USB Pro supported modes

Following a grid showing the IO modes supported by QLC+ for devices like DMX USB Pro and Pro Mk2.
If a mode is not listed here, it means it is not supported by QLC+ or the device itself because of hardware limitations, so please do not report them as issues in the QLC+ forum.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
DMX1 IN	o			o			o			o			o	
DMX1 OUT		o			o			o			o			o
DMX2 OUT (1)			o	o	o									
MIDI IN (2)						o	o	o				o	o	o
MIDI OUT (2)									o	o	o	o	o	o

(1) DMX2 OUT is available only on DMX USB Mk2 Pro
(2) MIDI IN and MIDI OUT are available only on DMX USB Mk2 Pro with a 5-way breakout cable.

MIDI OUT signals are sent from 1 to 512 as described in the MIDI plugin channels map

5 Tuning

Note: Manual tuning should never be performed except for some very particular cases. Use it at your own risk !
It is possible to change the DMX frame frequency for Enttec Open (and like) devices with a hidden settings key on each platform. The key tells QLC+ how many times each DMX frame (512 channels) should be sent to the universe per second. A value of "30" means 30 times per second (30Hz).
Please refer to the Manual parameters tuning DMX USB Enttec Open section

E1.31 input/output plugin

Introduction

QLC+ supports the E1.31 protocol (also known as sACN) through an input/output plugin that receives and transmits packets on the network.
No extra requirements are needed, since QLC+ has a native implementation of the E1.31 protocol that works on Linux, Windows and OSX systems.
The E1.31 plugin can send and receive packets from multiple network cards, virtual addresses and multiple universes per network card.
E1.31 packets will be sent on multicast addresses like this: 239.255.0.x, where 'x' is the universe number selected in QLC+.

Compatibility

QLC+ has been tested with the following E1.31 softwares/devices:

- DMXking eDMX2 TX - Output device

HID plugin

1.Introduction

The HID plugin supports the HID system on Windows and Linux.
HID is a generic way of mapping input/output devices such as joysticks, touchpads, keyboards, mice, etc.
The QLC+ HID plugin aims to support only joysticks and the FX5 USB DMX adapter.

2.Requirements

No special requirement is needed for this plugin. Just make sure your operating system is actually recognizing the device you're going to use. On Linux you might need some kernel modules (normally provided by modern distributions) and on Windows a driver provided by the device manufacturer.

3.Joysticks

QLC+ tries to detect the specific joystick functionalities, such as axes and buttons, as individual channels that can be mapped to your Virtual Console widgets.
Axes and buttons are mapped by QLC+ in sequential order, so for example if your joystick supports 2 axes and 4 buttons, they will appear in the input mapping dialogs like this:

- channel 1: X-axis
- channel 2: Y-axis
- channel 3: Button 1
- channel 4: Button 2
- channel 5: Button 3
- channel 6: Button 4

4.FX5 USB DMX

The FX5 USB DMX adapter is supported either for output and input DMX data. Check the product manual for the connection and peculiar issues.

MIDI input/output plugin

Introduction

This plugin gives input/output support for the MIDI protocol and gives the user freedom to control typical parameters like channels, Program Change and Control Change.
The MIDI plugin can be quite powerful used in combination with MIDI devices such as keyboards, MIDI controllers (like Behringer BCF2000 or KORG nanoKONTROL) or a software audio sequencer like Cubase or Ardour 3.
The usage can vary from fader-to-fader control (the BCF2000 case) to sequence triggering for synchronized shows (metronome controlled gigs using an audio sequencer)

Configuration

When you click on the configuration button a window will appear, showing all the MIDI input and output lines detected.
Each line has three options that can be changed depending on your needs:

- **MIDI Channel**: This is the channel where QLC+ will receive or send data through the MIDI system. MIDI channels can go from 1 to 16. The special "1-16" channel will tell QLC+ to receive or send data on any MIDI channel.
- **Mode**: This is the MIDI mode that QLC+ will use to send data through the MIDI system. This parameter can assume three possible values:
  - **Note velocity**: in this mode, QLC+ will send data using MIDI notes velocity. MIDI notes can go from 21 (A0) to 108 (C8) and each note can have a velocity going from 0 to 127, which will be doubled inside QLC+ to match the DMX range (0-255).
  - **Control Change**: this is one of the MIDI protocol messages (like Program Change) frequently used by MIDI controllers. Each device should list the supported CC messages in its user manual, so please consult it before using this mode. The CC range can go from 0 to 127 and can have values from 0 to 127, which will be doubled inside QLC+ to match the DMX range (0-255).
  - **Program Change**: this is one of the MIDI protocol messages (like Control Change) frequently used by MIDI controllers. Each device should list the supported PC messages in its user manual, so please consult it before using this mode. The PC range can go from 0 to 127 and can have values from 0 to 127, which will be doubled inside QLC+ to match the DMX range (0-255).
- **Initialization message**: This is a list of presets (templates) containing the intialization message that QLC+ will send when opening a MIDI device before using it. A detailed explanation of this functionality is written below.

Feedbacks

The MIDI plugin is the one of the QLC+ plugins supporting feedback. When QLC+ detects a MIDI device with an output line, it will enable the feedback check box in the Input/Output panel. Please note that output and feedback are exclusive, so they cannot both be used at the same time.
If your MIDI device supports a return channel, QLC+ can send a visual/mechanical feedback to it.

Devices such as the Behringer BCF2000 support this feature. This is very useful during live shows to have immediate knowledge of the current state of faders mapped in QLC+.

A small trick that can be achieved with QLC+ is to use feedback as a generic MIDI output line to trigger external controllers/sequencers.
Let's look at some examples:

- Input: **OSC** ---> Output: **DMX USB** --> Feedback: **MIDI**
- Input: **Enttec Wing** --> Output: **ArtNet** --> Feedback: **MIDI**

MIDI beat clock

Starting from version 4.5.0, QLC+ supports the MIDI beat clock
Not to be confused with the MIDI timecode, the MIDI beat clock is a useful signal to sync BPM-based devices such as a drum machine with your lights controlled by QLC+.
Two special MIDI channels have been mapped in QLC+ to control your Virtual Console widgets with a beat clock.
Here's a brief explanation of the special channels:

- **Channel 530**: A signal is sent on this channel when a beat clock starts or stops.
- **Channel 531**: This signal is sent every BPM. QLC+ doesn't take any notice of measures (e.g. 3/4, 4/4, 7/8), so when setting up your MIDI clock you need to consider how QLC+ will handle it.

Hint: If your controller is set to work at high BPM (e.g. 180-200), you might find difficult to catch the start signal. One trick for doing this is to catch the stop signal. Example:

1. Enable the QLC+ Virtual Console widget auto-detection
2. Hit play on your device generating the MIDI beat clock. QLC+ will detect channel 530 and will switch very quickly to 531
3. Stop the playback on your MIDI beat clock device. QLC+ will detect channel 530 again.
4. Disable the QLC+ Virtual Console widget auto-detection

In a similar way you can catch the beat signal as well. Just disable the auto-detection process before stopping the playback on your beat controller (invert steps 3 and 4).

MIDI initialization message

There might be cases where your MIDI device needs some commands to turn into a specific operating mode
The MIDI protocol can handle this through SysEx. These are particular messages to instruct a MIDI device how to behave.
QLC+ can use a XML template to achieve this that can be selected in the MIDI configuration panel.
Here's an example of how a template looks like:

```
<!DOCTYPE MidiTemplate>
<MidiTemplate>
 <Creator>
```

```
  <Author>Your name</Author>
 </Creator>
 <Description>A brief description of that the template does.</Description>
 <Name>Template name to be displayed by QLC+</Name>
 <InitMessage>F0 47 00 7B 60 00 04 41 09 00 05 F7</InitMessage>
</MidiTemplate>
```

You can create the ones you need and place them in your MidiTemplates folder.
You are welcome to submit them in the QLC+ forum.

QLC+ Channels map

To handle a mix of various MIDI messages (Notes, PC, CC, etc..), QLC+ remaps them into a sequential order.
Following, the channel numbers to be used in the Input Profile editor:

Channel	MIDI message
1	Control Change 1
...	...
128	Control Change 128
129	NOTE ON/NOTE OFF 1
...	...
256	NOTE ON/NOTE OFF 128
257	NOTE AFTERTOUCH 1
...	...
384	NOTE AFTERTOUCH 128
385	Program Change 1
...	...
512	Program Change 128
513	Channel Aftertouch
514	Pitch Wheel
530	MIDI Beat Clock: Start/Stop/Continue
531	MIDI Beat Clock: Beat

In OMNI mode, add 4096 * Channel number

OLA output plugin

Introduction

The OLA plugin allows direct communication between QLC+ and the OLA framework on the same machine.

Requirements

The OLA plugin requires OLA to be installed on the system.
Since OLA doesn't run on Windows, only Linux and OSX users can benefit from this plugin.
Information on how to download and install OLA can be found here.
QLC+ needs the OLA server to be running to be able to communicate with the OLA framework.
This can be done either manually by starting up "olad" from a terminal or in the configuration panel by ticking "Run standalone OLA daemon".

Configuration

When pressing the configuration button over a OLA output line, a small popup window will appear showing the basic information of how QLC+ outputs are mapped against OLA universes.
On the bottom, a check button will allow you to force the OLA server startup.

OLA Setup

When you have made sure that everything is working in QLC+ and have checked how the universes are mapped, you can setup OLA to output the signal received from QLC+ to a DMX device, either USB or over the network.
Here's an introduction of OLA usage.
Basically you need to open a web browser, connect to http://localhost:9090 or http://127.0.0.1:9090 and add a universe that has the same number mapped in QLC+ and select the desired output line.

OSC input/output plugin

Introduction

QLC+ supports the OSC protocol through an input plugin that receives packets from the network.

Requirements

The OSC protocol is available thanks to the liblo library, which is linked dynamically on Linux systems and statically on Windows and OSX.
In other words, Windows and OSX don't need any extra software, while Linux systems need the liblo library installed.

Configuration

The OSC plugin supports up to 4 inputs, one for each QLC+ universe. Each input has to listen to a unique port and this can be configured from the plugin configuration panel.
For each OSC network the following parameters can be set:

- **Port:** The port QLC+ will listen to in order to receive OSC data from your external controller. (Just enter the same port that your external OSC controller is using.)
- **Output address:** If you need to output DMX data through the OSC plugin, enter the address to send it to here (the address will also be used to send feedback). This must be specified in the form: *IPAddress:port*

Controllers

QLC+ has been tested with the following OSC controllers:

- TouchOSC. A pre-defined input profile is ready to use for the Mix16 layout.

Peperoni output plugin

1 Introduction

The Peperoni Output plugin supports USB-DMX output devices produced by Peperoni Light.

2 Requirements

2.1 Linux

On all Linux distributions, you need to install libusb. If you install QLC+ with the Ubuntu Software Center or some other automatic installer tool, this library will be installed automatically for you.

2.2 Mac OS X

Nothing special is needed for Mac OS X. All required components are already inside the Q Light Controller Plus application bundle because QLC+ uses the OS X native USB interface.

2.3 Windows

You must install the Peperoni USBDMX Windows drivers. Usually these come on a CD along with the Peperoni device you've bought.
If after installing the driver, QLC+ still doesn't detect your Peperoni device, copy the usbdmx.dll file you find inside the driver ZIP package (i386 folder) into the main QLC+ folder.

- Unzip the driver package to a folder on your hard disk.
- Plug the peperoni device to a USB port.
- If you're running a 32bit version of Windows, point the "Found new hardware" wizard to look for the driver under **windows/i386**.
- If you're running a 64bit version of Windows, point the "Found new hardware" wizard to look for the driver under **windows/ia64** or **windows/amd64**.

uDMX output plugin

1 Introduction

The uDMX Output plugin supports the Anyma uDMX USB-DMX interface on Linux and Mac OS X.

2 Requirements

2.1 Linux

On all Linux distributions, you need to install libusb. If you install QLC+ with the Ubuntu Software Center or some other automatic installer tool, this library will be installed automatically for you.

2.2 Mac OS X

On the Apple Mac OS X, you don't need any drivers at all since QLC+ uses the OS X native USB interface.

2.3 Windows

This device is not yet supported on Windows.

3 Tuning

It is possible to change the DMX frame frequency for all uDMX devices with a hidden settings key on each platform. The key tells QLC+ how many times each DMX frame (512 channels) should be sent to the universe per second. A value of "30" means 30 times per second (30Hz). Please refer to the Manual parameters tuning uDMX section

Velleman output plugin

1 Introduction

The Velleman Output plugin supports the Velleman K8062D interface on Windows operating systems.

2 Requirements

2.1 Windows

You must install the proprietary Velleman K8062D library to **C:\Windows\System32**.

Loopback input/output plugin -- EXPERIMENTAL

Introduction

Loopback plugin provides a way to control Virtual Console widgets from Scenes and other Functions. Data that QLC+ sends to the Output port is looped back to the Input port, where it can be used for external control. Obviously, the Output port and Input port may be attached to different universes.

The plugin provides 4 independent lines.

This plugin is mostly for advanced users -- unlike other input/output plugins, this plugin doesn't control any real device.

Configuration

Loopback plugin doesn't have any configuration. Simply attach Input and Output to desired universes.

Usage Examples

Buttons that set a (submaster or regular) slider to a predefined values

These button can have a fade time attached, these changes may be gradual.

Steps:

1. Set one universe for Loopback output (U1), and set the slider's input universe to Loopback Input (U2). Let's say the sliders external control is set to channel 14.
2. Add dimmer channel to U1 at address 14. In the channel modifiers dialog, set it to LTP (so that if the button goes off, the slider doesn't go to zero.
3. Create scenes with desired slider values for channel 14 in the U1 universe. You may set fade times as well.
4. Create buttons for the scenes

Fixture Definition Editor

Fixture Definition Editor is a separate application for creating and modifying fixture definitions used by QLC+. The definitions tell QLC+ (and users) important details about fixtures, such as which channel is used for pan movement, what value in which channel changes the beam color green, how the fixture is reset etc...

The main window in the Fixture Editor is just an empty workspace that contains the actual editor windows used to actually edit fixture definitions.

Important note: for many reasons, you SHOULD NOT save or copy your custom fixtures in the QLC+ system fixtures folder. The most important is that when you uninstall QLC+, the system fixtures folder gets deleted, so your fixtures.
You are recommended to save them in the user fixtures folder. To find it, please refer to the Q & A section of this documentation.

Controls

	Create a new fixture definition. Opens an empty Fixture Editor window.
	Open an existing fixture definiton. Open the fixture definition in a Fixture Editor window.
	Save the fixture definition in the currently active Fixture Editor window.
	Save the fixture definition with a given name in the currently active Fixture Editor window.

Capability Editor

With the Capability Editor you can define a value range for a certain functionality provided by a fixture channel. For example, if values 0-15 produce a white color capability, you might put:

- 0 to the **Min** field
- 15 to the **Max** field, and
- "White" to the **Description** field

Controls

Min	The minimum value of the capability
Max	The maximum value of the capability
Description	The name or description of the capability. Don't be too elaborate, just a few words will do.

Click & Go capability

The Click & Go functionality for a channel starts from the Capability Editor.
When a channels group is set to Gobo, Color or Effect, it is possible to assign the color or a picture that will be displayed by Click & Go on Scene Editor, Simple Desk and Virtual Console Slider.
This allows very quick visual feedback, and will avoid you having to remember the meaning of each range of values.
When a channel can support Click & Go, some additional controls will appear on the right side of the Capability Editor, displaying 3 butons and a preview box. Here's the meaning of each one:

Picture **...**	By clicking on this button you can select a picture for this range of values, to be displayed by Click & Go for visual association and quick access
Color	By clicking on this button you can select the first color for this range of values. When the first color is set, the 'Second color' button will be enabled. See below.
Second color	When clicking on this button you can select a second color for those cases where a fixture supports it. For example, this is quite useful for moving heads whose color wheels support intermediate positions.

Capability Wizard

Capability Wizard is a handy tool for creating multiple capability value ranges of same size. Usually this applies to fixed colors, gobo indices and various macro channels.

Controls

Start	The starting value for new capabilities. Sometimes there might be other capabilities at the start of the channel's value range that you can skip by adjusting this value.
Width	The size of each value range.
Amount	Number of capabilities to create.
Name	The common name for each capability. You can use the hash mark # to denote a place for an index number (i.e. "Gobo #" creates Gobo 1, Gobo 2, Gobo 3...)
Sample	Every time you change a parameter in the wizard, this list is updated to show you a sample of what kinds of capabilities will be created once you click OK.

Channel Editor

The Channel Editor is used to edit individual channels and the DMX value ranges of each capability (a green color, a certain gobo, prism rotation, etc..) that a fixture channel provides. Refer to your fixture's manual to get a detailed list of the fixture's channels and DMX values.

Controls

Name	The name of the channel
Group	The channel's logical group (its role in the fixture). • Assign a channel to the **Intensity** group if the channel should obey the HTP rule. Also, the Grand Master controls only Intensity channels by default. Note also when you create RGBAW/CMY channels, that they should be assigned to the *Intensity* group, and **NOT the Color group**. The Color Tool in Scene Editor is available only if a fixture provides **Intensity** channels for RGBAW/CMY, whose **Color** properties have been set accordingly (see the **Color** setting below). • Assign a channel to the **Color** group if the channel controls a fixed color wheel or pre-defined color macros. *Don't* assign individual RGBAW/CMY color channels to the Color group but instead use the **Intensity** group as described above The channel obeys the LTP rule. • Assign a channel to the **Gobo** group if the channel controls gobo wheel position or indexing. The channel obeys the LTP rule. • Assign a channel to the **Speed** group if the channel controls something related to speed (gobo rotation, rainbow speed, tracking speed). The channel obeys the LTP rule. • Assign a channel to the **Prism** group if the channel controls a prism. The channel obeys the LTP rule. • Assign a channel to the **Prism** group if the channel controls a prism. The channel obeys the LTP rule. • Assign a channel to the **Shutter** group if the channel controls a shutter or strobe. The channel obeys the LTP rule. • Assign a channel to the **Beam** group if the channel controls a beam shaper. The channel obeys the LTP rule. • Assign a channel to the **Effect** group if the channel controls something that doesn't quite fit into any of the other groups. The channel obeys the LTP rule. • Assign a channel to the **Maintenance** group if the channel contains fixture resetting or cooling fan controls or something similar. The channel obeys the

	LTP rule. • Assign a channel to the **Nothing** group if the channel should not appear in the Scene Editor or Fixture Console. • Assign pan channel(s) to the **Pan** group and respectively, tilt channel(s) to the **Tilt** group. See the **Control Byte** property below. The channel obeys the LTP rule.
Color	Only available when the channel has been assigned to the **Intensity** group. It indicates whether the channel controls a specific color out of the primary colors **Red, Green, Blue, Cyan, Magenta, Yellow, Amber, White or UV**. If the channel controls the fixture's master intensity, the **Generic** color setting should be used.
Control Byte	Applicable to channel pairs that make 16 bit values, usually **Pan** or **Tilt** group, but some newer fixtures support 16 bit dimmer or even RGB, gobo or focus channels. For 8 bit values (e.g. when the fixture supports only 8bit movement, only one channel for each movement), assign the **Coarse MSB** control byte to the channel. If, however, the fixture supports 16bit movement (two channels for each movement), or other 16 bit channel, you should assign the **Coarse MSB** byte to the channels that provide coarse value and the **Fine LSB** byte to the channels that provide fine value adjustment. If you are not sure, use **Coarse MSB**.
Capability list	Shows you a list of DMX value ranges for the currently edited channel. If a channel provides only one capability (for example pan or dimmer) the only thing needed in the list is one capability with a range of 0 to 255. For more elaborate capabilities, such as colors or gobos, you should create capability ranges for each of the colors (for example 0-15 white, 16-32 blue...)
	Add a new capability to the channel, using the Capability Editor.
	Remove the selected capabilities from the channel.
	Edit the selected capability, using the Capability Editor.
	Create new capabilities quickly with the Capability Wizard.

Fixture Editor

Fixture Editor windows contain everything needed to edit one Fixture Definition at a time. The windows are separated in three tabs: **General**, **Channels** and **Modes**.

Controls - General Tab

Manufacturer	The fixture's manufacturer name. For example "FooCompany".
Model	The fixture's model name. For example "FooZapper 2000".
Type	The fixture's generic type.

Controls - Channels Tab

The channels tab contains all possible channels that the fixture understands in all of its modes. The channel order doesn't matter in this tab at all. Instead, channels are arranged in certain order in each mode in the **Mode** tab. On the **Channel** tab, only the channel names, their **capabilities** (i.e. value ranges and their purpose) matter.

Channel list	Shows all of the channels that the fixture understand in all of its modes. Each channel item also shows all of its capabilities in a sub tree.
	Add a new channel to the fixture using the Channel Editor.
	Remove the selected channels from the fixture **and from all modes**.
	Edit the currently selected channel using the Channel Editor
	Copy the currently selected channel to the clipboard. Channels in the clipboard can also be pasted to other fixture definition windows.
	Paste a channel from clipboard to the fixture definition. Channels in the clipboard can also be pasted to other fixture definition windows.
	Open or close all of the channel nodes in the channel list.

Controls - Modes Tab

The modes tab contains all modes the fixture can be configured to.

Mode list	Displays all modes for the currently edited fixture. Each mode item can be opened to display the set and order of channels in that mode. • Name: The name of the mode (each name must be unique) • Channels: Number of channels in each mode

	Create a new mode for the fixture, using the Mode Editor.
	Remove the currently selected mode from the fixture. Removing a mode does not destroy any channels or other modes.
	Edit the currently selected mode, using the Mode Editor.
	Create a copy of the currently selected mode to the same fixture. Since modes are tightly coupled to a certain fixture's channels, modes cannot be copied across fixtures.
	Open or close all mode items.

Mode Editor

The Mode Editor is used to create and edit modes by picking sets of **Channels** in certain order (as defined by the fixture's manufacturer). Each editor window is divided into three tabs: Channels, Heads and Physical.

Controls - Channels Tab

In the Channels tab you can place the fixture's channels in an order that forms an actual representation of the DMX channels that the fixture understands when it has been configured in that particular mode.

Mode Name	The name of the mode (must be unique for each mode).
Channel list	Displays all channels present in the current mode in their proper order.
	Add channels from the fixture's channel collection to the mode. You can create/edit channels in the Fixture Editor's **Channels** tab. When you click on this button a new window will be displayed, showing two lists: • The list on the left shows the channels that have not yet been added to this mode. If you're creating a new mode, this list will show all the channels you created in the Fixture Editor's channels tab • The list on the right shows the channels that define the mode you're editing. If you're creating a new mode, this list will be empty. Please note that the order of the channels in this list is fundamental to the fixture mode definition. To move items from one list to another either use the central buttons or simply drag and drop them. When done, press OK to update the mode channel list.
	Remove the selected channels from the mode. Other modes' channel selections are left untouched.
	Raise the selected channel up by one.
	Lower the selected channel down by one.

Controls - Heads Tab

In the Heads tab you can define multiple heads for a single fixture, so that QLC+ knows to treat each of them individually in certain situations (such as Fixture Groups). If a fixture contains only one head when it's configured to use the currently edited mode (i.e. all of the mode's channels control one head at a time) there is no need to define the head. If, however, the fixture has multiple heads that you wish to be able to control, you must define each head here.

Heads don't have names as they can be thought of as simple "sub-fixtures" inside a fixture. Instead, they are automatically given index numbers. The ordering of the Heads should follow the physical fixture's configuration as closely as possible. So, if the heads go 1, 2, 3, 4 in the real world, don't define them as 4, 2, 1, 3 or something equally annoying.

Head list	Displays the list of heads currently defined for the fixture.
	Add a new head to the fixture, using the Head Editor.
	Remove the selected head from the fixture.
	Edit the selected head using the Head Editor.
	Raise the selected head upwards by one. You can use this to change the order of the heads within the fixture.
	Lower the selected head downwards by one. You can use this to change the order of the heads within the fixture.

Controls - Physical Tab

Because some fixtures allow their physical properties (e.g. pan/tilt range) to be altered between modes, it is beneficial that the fixture's actual physical properties are also tied to fixture modes.

Bulb	• Type: The type of the actual light source within the fixture. • Lumens: The light source's total luminous output in lumens. • Color Temperature (K): The light source's color temperature in Kelvins
Lens	• Name: The type/name of the lens, if applicable. • Min. degrees: The fixture's minimum beam angle in degrees. • Max. degrees: The fixture's maximum beam angle in degrees.
Technical	• Power Consumption: The fixture's total power consumption in Watts. • DMX Connector: The type of the fixture's DMX connector.
Dimensions	• Weight: The fixture's total weight in Kilograms. • Width: The fixture's total horizontal width in millimeters. • Height: The fixture's total vertical height in millimeters. • Depth: The fixture's total depth in millimeters.
Focus	• Type: The method of focusing the beam on different areas. For stationary fixtures this can be set to Fixed. • Pan Max Degrees: The maximum pan width in degrees. • Tilt Max Degrees: The maximum tilt height in degrees.

Head Editor

The Head Editor is used to create and edit heads by picking sets of **Channels** that are dedicated to a single head (as defined by the fixture's manufacturer).

Controls

The controls are very simple: place a checkmark on each channel that is used ONLY by the head you are currently editing. To add a head, see Mode Editor.
When adding heads in a fixture mode that provides a master dimmer channel, you can include that channel on each head. In this way a RGB matrix function will be able to control your heads out of the box, with no need to manual dim your matrices.

Channel list	Displays the list of all fixture channels available in the current mode. Channels that have been assigned to another head are disabled and cannot be selected because each channel can only belong to one head at a time.

Scene tutorial

This tutorial covers just the basic procedure of creating a fixture, a scene and a button and how to make them work. The point is to give the user a quick yet complete here-hold-my-hand-experience on how the Q Light Controller Plus is planned to work as a software for DMX lighting automation.

Before you start reading this tutorial, please familiarise yourself with the Main window and its parts.

Now we can really start!

Fixture Manager

Open the fixture manager panel now by clicking its icon.

The fixture manager is the heart of QLC+ fixture-oriented architecture. As its name implies, you can manage (add, remove and edit) your lighting fixtures from the fixture manager. On the left hand side of the manager there is a list that contains all of the fixtures in the current workspace (at the moment it will be empty). On the right hand side you can see some information related to the currently selected fixture. On the top of the fixture manager there is again another toolbar, containing the following buttons (from left to right):

Add new fixtures

Remove selected fixtures

Configure the selected fixture

Group a fixture selection

Ungroup a fixture from a group

Import a previously saved list of fixtures

Export a list of fixtures

Adding a fixture

Add a fixture to the workspace by clicking the add button.

On the left side of the dialog you can see a list of available fixture manufacturers. Each manufacturer's name is actually a folder containing a number of different fixture models produced by that manufacturer. You can find, for example, a "DJScan250" under the "Futurelight" folder. As you click a fixture from the list, you can see the Channels field on the right hand side change to display the number of DMX channels required by the selected fixture. There's also a list of the fixture's channels just under the Channels box.

Go ahead and select Futurelight DJScan250 but don't click OK just yet.

You can edit the new fixture's name in the Name field or you can stick to the default that the application suggests. If the fixture has different operational modes (different sets of channels), you can select one from the Mode box. The DJScan250 doesn't have more than one, so there's just "Mode 1" there. The fixture's DMX address can be set in the Address field and should be the same as the actual physical fixture's DMX address. The Universe field is used to assign the fixture to a physical DMX output universe. Usually each universe has its own cable coming from the PC.

If you wish to add multiple fixtures of the same type, you can increase the value in the Amount box. If you wish to leave some gaps between each fixture's address space, you can change the value in the Address gap box. **Let's leave these be for now.**

If you don't understand the DMX addressing principles, please consult your lighting equipment manuals for more information. In short, a DMX address is the first DMX channel of a particular fixture. In the case of a DJScan250 (which uses 6 channels), assigning for example 1 as its DMX address, reserves channels 1, 2, 3, 4, 5, and 6 for the fixture. The next fixture must then be assigned to DMX address 7 to prevent channel overlapping. **Let's use address 1 now.**

Back to the fixture manager

Click OK to close the dialog and add one Futurelight DJScan250 to the workspace.

On the left side of the fixture manager you can now see the fixture that we just added. On the right side, you can see the information about the fixture. You can edit the fixture's name, address and universe by clicking the Configure button. You can also change the fixture definition through the configuration dialog, but since we're quite happy with the fixture's information, we are not going go there.

My first function

Open the function manager by clicking its button on the main toolbar

If the fixture manager was the heart, function manager must be the brain of QLC+. With it, you can add, remove and edit various functions that perform the actual light automation for you. Let's look at the view first. There is, once again, a toolbar containing the following icons (from left to right):

Add a scene

Add a chaser

Add a sequence

Add an EFX
Add a collection
Add a RGB matrix
Add a script

Open the Functions wizard
Copy the selected functions
Destroy the selected functions

Below the toolbar, there is a list of functions within the current workspace. Since we haven't yet created any, it's still empty.

Scene editor

Add a new scene to the workspace by clicking the scene button

With the scene editor, you can create scene functions that basically contain values for a number of channels that relate to certain fixtures. On the left hand side of the editor there is a list of fixtures used in the scene, which is empty (but not for much longer). There are also some buttons to add/remove fixtures and enable/disable all the selected fixtures channels.
On the right hand side of the editor there is a list of channel groups used in the scene. Channel groups will not be used in this tutorial.

Add a fixture to the scene by clicking the add button. Select our one and only fixture, the DJScan250 from the list that pops up and click the OK button.

Now the scene has one fixture to control. Notice also that now there is new tab just beside the General tab with our fixture name. **Click the tab that says DJScan250.**

Now you can see a panel with 6 sliders & buttons, each of them representing a channel within the fixture, but all of the channels are disabled. Above each of the buttons (the ones with icons) there is yet another box which, when checked, enables the corresponding channel in the scene. If a box is not checked, that channel will not be touched by the scene at all. This is very useful when you wish to create for example a function that just sets the color of a scanner, without touching the gobo, intensity, pan, tilt and other features that you might wish to stay the way they are.

Set channels 3, 4 and 6 enabled by clicking on their check boxes.

You'll see that channels 3 and 4 changed their appearance and are no longer grayed out. You can also move their sliders and click their buttons. When you click on a button, you get a list of available capabilities that the fixture can do when a certain value is set to the channel. In the case of a DJScan250, channel 3 controls the color wheel, channel 4 controls gobo selection and channel 6 controls the fixture's shutter.

If you click the button on channel 3, you get a list of available colors that the fixture supports. Since many of these capabilities are specified by the manufacturer as a value range rather than a single value, many of these capabilities contain yet another sub-menu. Let's try setting a color.

Click the color button on channel 3, then move your cursor to "Orange" and select "80" from the sub-menu by clicking on that value.

Notice that the slider on channel 3 also moved up and the value above it now shows 80. If you have already patched an output plugin to the first universe, you may already see some action going on with your DMX equipment, since the scene editor also sends real DMX data to your fixtures as you edit the values. If you haven't done any mappings, don't feel bad, we'll get to it.

Next, click the gobo button and choose "Gobo 7", value 126 and then click the intensity button and choose "Shutter open", value 255.

Now we have a scene function that sets the value of DJScan250's channel 3 to 80, channel 4 to 126 and channel 6 to 255. Now all we need is a nice descriptive name for the scene. Click the

General tab to go to the general page where we started with the scene editor.

You can set a nice name to the scene by writing it to the Scene name edit box. **Type: "*DJScan250 Orange Gobo 7*" there.**

Now we're going to set a Fade In time to the Scene, so that when we play it it will fade to the values we set in a given amount of time.
Click on the icon from the Scene Editor toolbar. A tool will be displayed, allowing you to choose the Fade In and Fade Out times of the Scene. Let's change Fade In to 5 seconds. Either use the speed dial widget or manually write '5' on the second last text field where 0s is written. Close the Fade tool by clicking again on .

Back in the function manager

You can now see that the function manager displays a function called "DJScan250 Orange Gobo 7".
If you click the right mouse button over a function item, you get the same menu functionalities that are available in the upper part of the function manager. You can add new functions or edit existing ones. **But let's not edit this function anymore.**

Another function

Create another function just like the one you just made, but set the values for channels 3, 4 and 6 to 0 and name the function "DJScan250 Zero".

Virtual Console

Speaking of vital organs, we have already covered the heart and brain of QLC+, and we're only missing the body with its limbs to make the whole pack work. Well, so much for ridiculous analogies, let's move on and make our *"DJScan250"* fixture and its *"DJScan Orange Gobo 7"* function do some actual work for us.

You can close the function manager and the fixture manager now, if you want to make some

room but it's not necessary.

Click the virtual console button on the main window to show the virtual console tab.

Creating a button

At first, the virtual console is just an empty window without much to look at. There's a toolbar at the top of the panel, with icons to Add new widgets, Edit for editing widget properties and Tools for various tools to control the virtual console behaviour. You can also click the right mouse button on any virtual console widget to access a menu that contains most (but not all) of the options that are accessible through the menu bar.

Click the icon to add a new Button widget.

An empty button appears to the virtual console.

Attaching a function to the button

Double click on the button or on the icon to bring up the button's configuration dialog.

From this dialog you can edit the button's properties:

* Set the Button label that is displayed on the button
* Attach a Function to the button
* Bind a Key combination to act as button presses * Bind the button to an External Input source
* Set the Button press behaviour

Click the attach button to open a function selection dialog. Double-click on the *"DJScan250 Orange Gobo 7"* function to attach it to this button.

We don't necessarily need a name for the button, but if you feel like giving it one, please do.

Click OK to accept these changes and close the dialog.

A little fine-tuning

If you gave the button some name, you'll notice that it doesn't quite fit and gets trimmed to a rather short version of the original (unless you gave it a two-letter name). You can resize the button to any size you like by grabbing it from the box on the button's lower-right-hand corner and dragging the button a little bigger. But hey, let's change the button's color now.

Click on the button again, and then click on the icon. Select an orange-tinted color from this dialog and click OK.

Now the button should have an orange background color. Move the button a bit to the side so that the next button won't appear on top of it. Well, there's no harm in that, it's just an inconvenience - you want to be able to see both buttons, do you not?

Another button

Create another button just like the first one, but attach the function *"DJScan250 Zero"* to the second button and set the button background color to black and foreground (text) color to white.

Seeing the results

Since we haven't covered DMX output patching at all, you probably have a dummy output plugin assigned to all output universes and you can't get any real DMX output from your computer. This is OK for now. If you're interested in output mapping details, refer to the Output mapping howto.

Click the monitor button on the main application toolbar to bring up the DMX monitor window.

You should see a bunch of numbers, and the name of our fixture "DJScan250" inside a bar over numbers 001 - 006. These numbers represent DMX channels and the values below them represent those channels' values. Since we're operating on with a dummy output plugin, the monitor is all that we see for now.

Click the mode switch button on the right-hand corner of the main toolbar to switch to operate mode.

Hold your breath...

Pay close attention to the monitor while you click the buttons on the virtual console (you know, the one we just created). Do you see some running numbers that gradually go towards 80 on channel 3, 126 on channel 4 and 255 on channel 6? Nice.

Note that if you click both buttons simultaneously, the result is usually far from what is wanted. You need to stop the previous function by clicking its button once more (so that the button flashes and stays up) to stop the function and then start the other function.

Multipage frame tutorial

This tutorial covers how to set up a multipage frame and related remote MIDI control. The main advatages of multipage frames are better usage of screen estate, and more functions can be controlled by limited number of available real faders.

I will use Behringer BCF2000 as a MIDI controller in this tutorial, because that's what I own, but you can use any suitable controller. Besides that BCF2000 has flying (motorized) faders that are very handy when switching pages.

Note: In previous QLC+ versions, it was require to edit MIDI profile to define Previous page and Next page buttons. This is no longer needed.

Executive summary

If you don't want to read the whole text, here are the bare bones of it:

- Create frame/soloframe, enable pages in its properties, enter number of pages, and set the buttons from previous step as external input. Now page switching should work.
- Put widgets to pages, external control will have "(Page X)" appended.
- That's it.

The plan

We will proceed with this tutorial in several steps:

- Initial setup
- Create multipage frame
- Setup remote control
- Useful tips and improvements

Initial setup

To make thinds easy, let's assume we want to control 32 PAR cans on a dimmer addressed 1 to 32. we want to control them from BCF2000 MIDI controller (that has 8 flying faders). We want to make one fram with 4 pages, each containing 8 sliders one for each PAR can. If you don't have enough dimmer channels at hand, either skip some channels and use only 1-2 on each page, or just check the output in the DMX monitor window.

First, go to Fixture Manager and add 32 generic dimmer channels.

Next things will happen in Virtual console .

Create Multipage frame

The most basic way how to create multipage frame is to create simple frame . Now right-click on the frame to open context menu, and select "Widget properties" (or press CTRL-E). Go to "Pages" tab. Select "Enable", to enable the multipage functionality. Enter "4" in the "Number of pages" edit box. Press "OK".

Now we'll add the faders for the first page. Add 8 faders one by one. Then click on the button and do the same for page 2, and again for pages 3 and 4. For each button, open its Widget properties, select level mode, and select respective dimmer channel.

Now change to operate mode, and check if the sliders work.

Alternative approach

Instead of separately creating sliders for each page, we could create them *before* enabling pages and we would select also "clone first page widgets". Then all sliders will be cloned to all pages.

MIDI control

In the next step, we are going to setup MIDI remote control. In other words, we want to control switching of the pages and the sliders from the MIDI controller.

We start in Input/Outputs tab and setting basic MIDI control. Select e.g. universe 2 for MIDI control, choose BCF2000 MIDI 1 port; check both "input" and "feedback", and select BCF2000 profile.

Now go to the virtual console, right click the multipage frame, and select "Widget Properties". Select some buttons as external controls for Previous and Next page, either by autodetection, or choose from the list (Don't forget to disable autodetect for Previous Page when you are setting Next Page, otherwise you will change both!). I use "Button 9 Top" for previous page, and "Button 10 TOP" for next page, but you can choose any buttons. Push OK to close the dialog. Now the two buttons should work. Try them now to switch the pages.

The last thing to do is to link the sliders in the frame with the BCF faders. Select page 1, and open properties for every slider on the page, setting external control as usual. The BCF sliders will have "(Page 1)" appended to them ("82: Slider 1 (Page 1)"). When you are finished with the first page, select page 2, and do the same, and then page 3 and 4. Since with BCF200 we have MIDI feedback, it's not important if we switch pages in QLC+ or on the MIDI controller. For controllers without feedback, it's crucial to use buttons on the controller, otherwise the pages will get out of sync.

Now you can test the sliders from MIDI controller - all of the 32 PAR cans should be controllable using those 8 faders and 2 buttons.

Tips and tricks

Multipage solo frames

Solo frames work across pages. If you press a button on any page, buttons/widgets on all pages will et released.

It is not possible (yet?) to set some buttons to select concrete page, e.g. button to select Page 1, another for Page 2, etc.

Sound Control Tutorial

This tutorial covers how to set up sound control for chasers.

The steps

1. In Function Manager, create Chaser, set steps to have infinite hold time
2. In the Virtual Console, create cue list for that chaser
3. Add audio triggers widget
4. In the audio trigger properties: choose proper frequency band (or use volume band), set type to VC Widget and select your cue list. Audio trigger will press "next" button on the cuelist.
5. now adjust parameters:
   - enable threshold - when the volume goes over enable threshold, the button will be pushed.
   - disable threshold - when the button is pushed, it won't be pushed again before the volume goes below this level. This is so that the button is pushed only once per beat
   - divisor - the button will be pushed every x-th beat - when you want to advance the chaser every other beat, put here 2, every 4-th beat - put 4, etc.
6. close properties, start Operate mode, enable audiotriggers and that's it.

The whole process is described in this video.

Audio Trigger widget offers more ways of sound control. For details, see the description.

BCF2000 and LC2412

This is small howto to setup remote control with Behringer LC2412 connected through BCF2000. Everything said here is also valid for BCR2000.

With this setup (BCF2000 + LC2412) we get:

- 8 motorized faders
- 8 turn encoders
- 30 non-motorized faders
- lots of buttons

That makes a pretty nice console and a much less clicks with mouse :)

Now the steps:

1. Connect MIDI cable from LC2412 MIDI out to MIDI IN on BCF2000. Connect USB cable from BCF2000 to the computer.
2. Set BCF2000 to U-2 mode
    - Press and hold EDIT and then push STORE.
    - Release both buttons.
    - Turn the leftmost encoder until it shows u-2.
    - Push EXIT button.
3. Start QLC+
4. In the INPUT/OUTPUTs tab:
    - For one universe, choose BCF2000 MIDI 1 port; check both "input" and "feedback", and select BCF2000 profile
    - For another universe, choose BCF2000 MIDI 2 port; check "input" only - LC2412 does not have feedback capability. Choose LC2412 profile.
5. Now you can choose from knobs, faders and buttons of either device for your VC controls.

Command-Line Parameters

QLC+ supports a number of command line parameters to automate/extend some functionalities on startup.
Using command line parameters can be tricky depending on the operating system you're using:

- **Linux**: just open a terminal and type "qlcplus" followed by the parameters you need
- **Windows**: create a shortcut of qlcplus.exe (usually located in C:\QLC+) on your desktop. Right click on the shortcut and select "Properties". In the "Target" field you will see something like "C:\QLC+\qlcplus.exe". There you can add command line parameters. When done click OK.

  Example: Setting Czech locale:

- **OSX**: This is the most difficult case since QLC+ on OSX is bundled into a DMG package. You need to open a terminal and "cd" into the QLC+ DMG like this:
  cd QLC+.app\Contents\MacOS
  When done, type "qlcplus" followed by the parameters you need

-c or --closebutton

Description: Define a position for a close button in the virtual console. The button can be used

to close QLC+. Only has an effect in kiosk mode (see -k) and is most useful when in fullscreen mode without a window manager.

Parameters: x,y,w,h

Examples:
Create a button at (x400, y500) whose size is (w70, h50):
qlcplus -c 400,500,70,50
qlcplus --closebutton 400,500,70,50

-d or --debug

Description: Enable debug mode and optionally set the output level. Note that since 4.8.0 messages for level DEBUG (0) are not included in released binaries.

Parameters: Level (see QtMsgType)

Examples:
Enable debug mode and set debug level to 0:
qlcplus -d
qlcplus --debug

Enable debug mode and set debug level to 3:
qlcplus -d 3

-f or --fullscreen

Description: Start the application in fullscreen mode

Parameters: Method (either 'normal' or 'resize')

Examples:
Tell the window manager to give the whole screen space to QLC+:
qlcplus -f
qlcplus --fullscreen
qlcplus -f normal
qlcplus --fullscreen normal

Resize QLC+ to take up the whole screen space (useful in custom X11 sessions without a window manager):
qlcplus -f resize
qlcplus --fullscreen resize

-g or --log

Description: Log debug messages to a file ($HOME/QLC+.log)

Parameters: None

Examples:
Enable debug messages and store them to log file:
qlcplus -d 0 -g
qlcplus --debug 0 --log

-h or --help

Description: Display command-line help (only in Linux & OS X)

Parameters: None

Examples:
Display the command-line help:
qlcplus -h
qlcplus --help

-k or --kiosk

Description: Enable kiosk-mode (only virtual console is visible and the application is locked in operate mode)

Parameters: None

Examples:
Start the application in kiosk mode:
qlcplus -k
qlcplus --kiosk

-l or --locale

Description: Use the given locale for translation

Parameters: Locale name (currently supported: ca_ES, cz_CZ, de_DE, en_GB, es_ES, fi_FI, fr_FR, it_IT, ja_JP, nl_NL, pt_BR)

Examples:
Use finnish language:
qlcplus -l fi_FI
qlcplus --locale fi_FI

-o or --open

Description: Open the given workspace file

Parameters: File name

Examples:
Open mydesk.qxw:
qlcplus -o mydesk.qxw
qlcplus --open mydesk.qxw

-p or --operate

Description: Start the application in Operate mode.

Parameters: None

Examples:

Start QLC+ in operate mode:
qlcplus -p
qlcplus --operate

-v or --version

Description: Display the current application version number

Parameters: None

Examples:
qlcplus -v
qlcplus --version

-w or --web

Description: Enable remote web access on port 9999

Parameters: None

Examples:
qlcplus -w
qlcplus --web

Kiosk Mode

QLC+ has a special mode when your lighting needs to be operated by inexperienced people, or you just want to prevent unwanted changes. It's called the Kiosk mode. In this mode only the Virtual Console is displayed, so no editing is possible.

When you want to start QLC+ in kiosk mode, run:

```
qlcplus -k -f -o workspace.qxw -p -c 500,10,32,32
```

- **-k** enables the kiosk mode
- **-f** puts QLC+ in fullscreen mode
- **-o workspace.qxw** opens specified workspace
- **-p** starts operate mode
- **-c 500,10,32,32** creates a close button with size of 32x32 pixels, 500 pixels from left and 10 pixels from top.

You can of course specify your own workspace file for **-o**, and size/position for **-c**.

You can omit any of the paramteres if you need to - for example, omit **-c 500,10,32,32** if you don't want close button, and omit **-f** if you don't want fullscreen.

All command-line parameters are described here.

Web Interface

By default, QLC+ includes a web interface based on the Mongoose project HTTP server.
It comes very handy when you need to run QLC+ on a device without a display (headless system) either to work standalone or for remote controlling.
The web interface is not enabled by default but can be easily activated by running QLC+ with "-w" or "--web" option. Please refer to the command line paramters page of this documentation to learn how to do it.

The web interface can be accessed from any modern web browser running on any device, such as a computer, a tablet or a smartphone. Your browser need to support the web sockets technology to communicate with QLC+.
It is recommended to use Google Chrome.

To access the QLC+ web interface simply connect to this address:
http://*IP address*:9999

Where *IP address* is the IP address of the device you want to access via web. For example:
http://192.168.0.100:9999
The web interface consists in two pages:

- Virtual Console
- Configuration

Virtual Console page

This page is displayed by default when accessing the web interface address and it represents the QLC+ Virtual Console.
If a project is loaded, this page will display the widgets previously created with QLC+, otherwise it will just show an empty page.
It is possible to load a project with the **Load project** button placed on the top left corner of the page. A window will show up, allowing you to choose a file from the device you're using to control the device where QLC+ is running.
The project file will be transferred via web and loaded by QLC+.
To access the QLC+ configuration page, just click on the **Configuration** button.

Configuration page

This page allows to remotely set the QLC+ configuration, divided in three areas:

- **Universes configuration**: Allows to set the inputs, outputs, feedback, profiles and passthrough mode for each QLC+ universe. This is basically the same functionality of the QLC+ input/output panel.
  Since a QLC+ project contains also the I/O information, most likely you won't have to manually configure it on this page, but just check if everything is correct.
- **Audio configuration**: Allows to select the audio devices used for audio playback or audio input.
- **User loaded fixtures**: Allows to remotely load a custom fixture to QLC+.
  When clicking on the **Load fixture** button, a window will show up, allowing you to choose a

file from the device you're using to control the device where QLC+ is running.
The fixture file will be transferred via web and loaded by QLC+.
When adding new custom fixtures it is recommended to reload a project or either restart QLC+ on the target device.

Once the configuration has been set, it is possible to go back to the web interface main page by clicking on the **Back** button, placed at the top left corner of the page.

Parameters tuning

This section explains how to manually get at and tune some QLC+ parameters not available from the UI.
Keep in mind that if you cannot change them from the UI, it means there is a good reason for it.

Warning: DO NOT edit the configuration files manually unless you know what you're doing. Any misplaced change can cause program crashes or awkward instability.

1 QLC+ configuration location

Linux

Configuration files are located in your user $HOME directory, in the .config/qlcplus folder.
Here's the quick command to access it from a terminal:
`cd $HOME/.config/qlcplus`
You will find both QLC+ and Fixture editor configuration files.

Windows

Configuration parameters are stored in the Windows registry.
To access it, run the "regedit" tool and search for the key named "qlcplus".

Mac OSX

Configuration files are located in your user $HOME directory, in the Library/Preferences folder.
Here's the quick command to access it from a terminal:
`cd $HOME/Library/Preferences`
QLC+ configuration file is called `net.sf.Q Light Controller Plus.plist` while the Fixture editor configuration file is called `net.sf.Fixture Definition Editor.plist`.

2 Configuration Reset

Sometimes it might be necessary to reset the QLC+ configuration and bring QLC+ to a "Factory defaults" state.
To do so, locate the configuration as explained in the first paragraph, then do the following:

- On Linux and OSX use the `rm 'filename'` command to delete the configuration file
- On Windows delete the whole 'qlcplus' KEY using regedit

3 Parameters syntax

Due to Qt differences on different platforms, parameters are stored in different ways depending on your Operating System.

Linux

Parameters are stored in a plain text file that you can modify with a plain text editor like gedit, kwrite, kate, nano, vim or similar. They are presented as follows:

```
[category]
name=value
```

Windows

Parameters are stored in the Windows registry and can be modified with the regedit tool included in every Windows version. They are presented as follows:

```
"category" is displayed as a folder
"parameter" is contained in "category" and is represented as a key. It is most likely that all QLC+ keys are an integer type.
"value" is the actual value of "parameter"
```

Mac OSX

Parameters are stored in a plain text file that you can modify with a plain text editor like TextWrangler or similar. They are presented as follows:

```
<key>category/name</key>
```

```
<string>value</string>
```

4 Parameters list

Most likely you will not find the following parameters in a standard QLC+ configuration. To make them effective, you need to **add** them at the end of your configuration file.

4.1 Simple Desk

Category: simpledesk
Name: channelsperpage
Type: integer
Description: Set the number of DMX channel sliders displayed per page
Default: 32

Category: simpledesk
Name: playbacksperpage
Type: integer
Description: set the number of playback sliders displayed per page
Default: 15

4.2 DMX USB Enttec Open frequency

Category: enttecdmxusbopen
Name: frequency
Type: integer
Description: Set the DMX frame frequency in Hertz for Enttec Open (and similar) devices
Default: 30

4.3 DMX USB Enttec Open channels number

Category: enttecdmxusbopen
Name: channels
Type: integer
Description: Set the maximum number of DMX channels transmitted by an Enttec Open (and similar) devices. This could solve flickering issues for some cases. For example you might try with 256.
Default: 512

4.4 uDMX frequency

Category: udmx
Name: frequency
Type: integer
Description: Set the DMX frame frequency in Hertz for uDMX devices
Default: 30

4.5 Master Timer frequency

Category: mastertimer
Name: frequency
Type: integer
Description: Set the QLC+ core timer frequency in Hertz
Default: 50

GUI style customization

Starting from version 4.5.0, QLC+ can read a user custom file to change the GUI appearance in a very accessible way.
If no file is found, QLC+ will start with the default style.

The GUI style file

The file name is hardcoded into QLC+ and must be: `qlcplusStyle.qss`
The style file must also be placed in a specific path which is:

- **Linux**: $HOME/.qlcplus
- **Windows**: Your user folder/QLC+
- **OSX**: Library/Application Support/QLC+

The style file must have a CSS syntax. If you're comfortable with web designing, you should find the creation of this file very easy !
Since the style file is strictly related to the inner Qt objects, you might want to read the following articles to find out the elements' names and the additional CSS properties the Qt team added to the default CSS syntax.
Qt Style Sheets
Qt Style Sheets Examples
It is up to your imagination how you prefer to customize the QLC+ GUI appearance! If you find a style worth sharing, don't forget to send in your contribution by posting it online in the QLC+ forum

QLC+ dark blue style example

Just to give you an example of how easy this process is, here's a blue-ish dark style for QLC+.
If you copy the following lines into `qlcplusStyle.qss` in the right location as explained above, you will see it immediately applied to QLC+.

```
QMainWindow, QDialog
{
 background-color: #404B57;
 color: #E6E6E6;
}

QTreeWidget
{
 background-color: #3A444F;
 alternate-background-color: #404B57;
 color: #E6E6E6;
}

QTextBrowser
{
 background-color: #3A444F;
 color: #E6E6E6;
}
```