QLab 3

Reference Manual

Last updated on January 19, 2018 for QLab 3.2.12

Table of Contents

Chapter 7: How To

Chapter 1: General

New in v3

General Improvements

Better, faster, stronger.

We have rebuilt the audio and video engines from scratch, and they're better than ever.

Redesigned interface.

QLab now features a darker interface designed to be easy on the eyes in a dim theater.

OSC control.

QLab can send and receive Open Sound Control commands. Learn about this extensible command language at http://opensoundcontrol.org

Show mode and edit mode.

Safeguard against accidental editing during a show; rather than simply locking a workspace, show mode now hides the inspector and prevents access to all edit features.

Broken cues list.

A new icon now appears in workspaces containing broken cues. Clicking it provides a summary of all broken cues with explanations of why they are broken.

Cue flagging.

Flagged cues appear in the broken cue summary as a convenient tool for keeping track of cues that require further attention.

A graceful way to panic.

Fade out a single cue or all currently playing cues over a user-defined “panic duration”: a new, smoother alternative to a hard stop.

Automatic cue firing.

Select a cue to automatically fire when a workspace is opened.

A smarter wall clock trigger.

The wall clock trigger now includes a day(s) of the week option.

Audio

AudioUnits.

Insert AudioUnits on cues, cue outputs, and device outputs, and target AU parameters using Fade cues.

Mic cues.

A new type of cue allows you to bring in audio from any input on an audio interface and route it like an Audio cue.

Slice, loop, devamp, repeat.

Audio and Video cues are now loopable in user-defined slices for better control over cue looping, allowing you to create more intricate sequences of looping and devamping in a single cue.

Variable rate playback.

Play cues at variable rates ranging from .03 to 33 times the regular rate, with or without pitch shifting.

Playback rate fading.

Control the rate of a playing cue live using a Fade cue.

Negative load to time.

Load to time using a positive value to begin a cue at a designated time from the start of the cue, or use a negative value to begin a cue at a designated time from the end of the cue.

Control buffer size.

QLab can now control the buffer size for any connected audio interface.

Video

On the surface.

Create a surface of any size for video output and assign one or more screens (projectors, LED walls, monitors, or other image displaying devices) to the surface.

Edge blending.

If more than one screen is assigned to a surface, QLab can automatically determine how much they should overlap and blend them to create a single seamless image.

Corner pinning.

Each screen in a surface can be independently perspective-corrected by dragging the corners of the screen to compensate for your projection angle.

Built-in masking.

Mask selected areas of each surface to create irregularly-shaped surfaces. In QLab 3.0, masked out portions are opaque. In QLab 3.1, masked out portions are clear to allow multiple surfaces to be stacked.

Audition window.

Any cue fired while the audition window is showing will play in the audition window rather than through your actual outputs. The audition window can have its own audio patch so you can preview cues in the privacy of your own ears.

Your very own thumbnail image.

The inspector view will display a thumbnail image of a user's actual video rather than a default image.

Rotate videos in 3D.

3D quaternion rotation guarantees smoother movement than XYZ rotation. You can also rotate around a specific axis if you so desire.

Variable rate playback.

Control the rate of Video cues just like Audio cues.

Go directly to the bottom, do not pass “Go,” do not collect $200.

In addition to sending a cue directly to the top layer, you can now assign a cue to the bottom layer, causing it to be displayed behind other currently playing cues. The most recently fired cue will take precedence.

Relative fades.

Rotate a video to an absolute value or to a value relative to its current state.

Other

QLab Remote.

QLab Remote is an iOS App available in the App Store. In order to be as useful as possible to users, we have given it a small set of features to start and add more continually based on your feedback.

QLab command line controller.

qcmd is an OSC-based command line client for QLab 3 available at http://figure53.com/code/

System Recommendations

Overview

QLab 3 is a Mac-only program. It runs on any Mac that is running macOS 10.8 through macOS 10.14.6. QLab 3 is not compatible with macOS 10.15 or later.

Because of QLab's great flexibility and the varied scenarios in which it is used, it can be difficult to determine ahead of time how much computer power a given QLab workspace will require. What follows is a discussion of general concepts surrounding processor, GPU, RAM, and hard disk use for QLab. Please take this information not as a firm set of instructions about what to do, but rather as a set of recommendations about what to consider.

QLab is not supported on "hackintoshes" at all. Please do yourself a big favor, and just keep away from those.

Processor

QLab 3 has been run on Core 2 Duo and newer processors. The more work QLab needs to do, the happier it will be with a more powerful processor. Large numbers of audio or video cues playing back simultaneously, for example, benefit from an i7 processor and its improved handling of multithreading tasks.

GPU and VRAM

For audio-only users, GPU considerations are probably negligible. For video folks, what you need depends entirely upon what you're trying to accomplish. A late-2012 Mac Mini can drive two HD displays simultaneously; one for your operator and one for your projector. Since those two displays share a single GPU, you can improve overall performance by lowering the resolution on your operator's display; fixed GPU power doing less work for the operator's display means more power available for video crunching. If you use a Mac Pro "cheese grater" tower (that is, pre-2014), dedicating one modest video card for your operator display and one higher-end card for each projector, or one card per two projectors, is a good strategy. For the newer "trash can" Mac Pro, all graphics connections are on the same GPU, which is much more powerful than its predecessors. Testing is, as always, important.

RAM

Loading and playing cues uses RAM, so the more audio or video that needs to be loaded at any given moment, the higher the RAM requirement will be. 4 GB is a nice minimum amount of RAM to work with, and is conveniently the minimum amount of RAM offered with any new Mac. As with processing power, complex shows can benefit from (and may require) more RAM. QLab 3 is able to address as much RAM as your Mac provides.

For Macs with an integrated GPU, which is all Mac Minis, the MacBook, all MacBook Airs, some iMac models, and some MacBook Pro models, the GPU uses a portion of system RAM as VRAM. The size of this portion is based on the total amount of system RAM installed, so the more RAM you have, the more of it will be used for VRAM. While we don't recommend using a Mac with an integrated GPU for video-intensive shows, if you do use such a Mac we strongly encourage you to install the maximum possible amount of RAM.

Hard disk

QLab is happiest with either a 7200 RPM hard drive or a solid state drive. We do not recommend using a 5400 RPM drive for anything other than the simplest shows. On the other side of the spectrum, solid state drives (SSDs) are, to put it plainly, really really fast. The more data you're pulling off your disk and pushing out of your speakers or projectors, the more an SSD is a good idea. Note that Apple's Fusion Drive option, though it technically includes an SSD, is not recommended for use on a show computer under any circumstances, as the user has no control over which data goes to the SSD versus the HD, nor any say in when the OS decides to shuffle data between the two.

Video Output

The best way to output video from QLab is to use the built-in video connections on your Mac (including PCI cards on "cheese-grater" Mac Pros.) Using a Pro Video or Pro Bundle license, you can also output video directly to Blackmagic Design's DeckLink, Intensity, and UltraStudio devices, although this comes at the cost of a reduction in CPU performance and an increase in latency.

We do not recommend, nor do we support, video output via DisplayLink USB-connected monitors and video adapters. While these devices can work, they are not GPU accelerated and their drivers do not have a solid history. Your mileage may vary, but Figure 53's position is to avoid using these devices entirely.

Ask Us

If you have specific questions about hardware requirements, please email the support team with your show's needs and we will be happy to advise you.

Preparing Your Mac

There are a number of programs, processes, and tasks that your Mac runs either periodically or all the time in the background by default. Many of these programs are essential, but many are not and disabling them will increase the total percentage of your computer's resources which are available to QLab.

What follows here is a list of these programs or processes which we recommend disabling, and instructions for doing so. This section presupposes a basic understanding of the Mac OS and at least a passing familiarity with the Terminal.

Note: A version of this information was published on our blog in an article entitled *Prepare, Execute, Troubleshoot*.

A Computer Prepares

Disable Spotlight

Spotlight periodically updates its index of all files on all attached disks, and this updating can cause the disk to be momentarily unavailable to QLab. This can cause late cues or stuttering in playback. To prevent Spotlight from updating its index, open a Terminal window and enter this command:

```
sudo mdutil -a -i off
```

Disable Display Sleep, Disk Spindown, and System Sleep

Obviously we don't want our computer going to sleep during a show. The Mac OS has independent sleep intervals for the display, the hard disk, and the whole system. To prevent all three kinds of sleeping, open a Terminal window and enter this command:

```
sudo pmset -a displaysleep 0 disksleep 0 sleep 0
```

Disable Screen Saver

Likewise, we don't want the screensaver coming up, particularly if QLab is running video. To prevent that from happening, open a Terminal window and enter this command:

```
defaults -currentHost write com.apple.screensaver idleTime 0
```

Disable Time Machine

Backups are wonderful. You should back up everything, all the time. But on a computer used for your show, backups should only be done manually. Time Machine, much like Spotlight, uses indexing and background processes which can take hold of the disk at inopportune moments. To shut off Time Machine, open a Terminal window and enter this command:

```
sudo tmutil disable
```

Disable Software Update

You don't want your computer trying to update software in the middle of a run, let alone in the middle of a performance. To disable Software Update, open a Terminal window

and enter this command:

```
sudo softwareupdate --schedule off
```

Disable Dashboard

Dashboard, largely neglected by Apple these days, is a pernicious little vampire of CPU time and network access. Also, if accidentally invoked, it takes over the screen of your Mac entirely, which can be surprising and confusing and lead to missed cues. To disable Dashboard entirely, open a Terminal window and enter this command:

```
defaults write com.apple.dashboard mcx-disabled -boolean YES
```

Stay Off The Internet

Many individual applications, including QLab, have their own internal scheme to check for updates. You can turn them off manually, and we recommend that. But the best way to guarantee that automatic software updates or any other network traffic won't bother your show is to disconnect the show computer from the Internet. We strongly encourage this. If you use a network to connect your QLab computer to other hardware, and your show doesn't require Internet access, make sure that network is a closed LAN (local area network) and has no path to the Internet.

Log Out of iCloud

Even when your Mac is offline, iCloud is surprisingly assertive about checking in and trying to phone home. Logging out of iCloud ensures that this check-in process doesn't

claim processor power when you need it.

```
Open System Preferences;
Choose iCloud;
Click "Sign Out".
```

Minimize Internet Accounts

Similarly, any accounts used to sync Mail, Contacts, and Calendars can potentially try to access the Internet and take up processing power while doing so, even while network access is disabled.

```
Open System Preferences;
Choose Internet Accounts;
Choose an account;
Uncheck each service type;
Repeat for each account.
```

Disable Hot Corners

Perhaps a lesser danger to a smooth running show, hot corners are nevertheless potentially problematic and we like to disable them, particularly when using screen sharing or VNC. To disable all four hot corners, open a Terminal window and enter these four commands, one at a time:

```
defaults write com.apple.dock wvous-tl-corner -int 1
defaults write com.apple.dock wvous-bl-corner -int 1
defaults write com.apple.dock wvous-tr-corner -int 1
defaults write com.apple.dock wvous-br-corner -int 1
```

Disable Notification Center

Depending on the way that you're using QLab, the iOS-style notifications system in Mac OS 10.8 and newer can be anything between a minor nuisance to a seriously embarrassing accidental component of your projection design. To prevent those charming little pop-up bubbles in the top right corner of your screen, open a Terminal window and enter this command:

```
launchctl unload -w /System/Library/LaunchAgents/com.apple.notificationcenterui.plist
```

Disable Mission Control

Mission Control is the feature that shows you all the open windows of all the open applications on your Mac, and makes it easy to move between them. On most laptops, the keyboard shortcut for this is F3, which is marked with three little rectangles. Needless to say, accidentally invoking Mission Control, particularly in a show with projections, can be a problem. To keep your own control of your theatrical mission, open a Terminal window and enter this command:

```
defaults write com.apple.dock mcx-expose-disabled -bool TRUE
```

Restart the Dock

Oddly, the Dock is in control of several of the system components that we just adjusted. Restarting the Dock allows these changes to take effect. Open a Terminal window and enter this command:

```
killall Dock
```

And If You're Doing Video...

If you're using QLab for video, there are two more critical settings:

Disable *Mirror Displays*

When you have more than one display connected to a Mac (including the built-in display on a laptop or iMac), you can either have the displays mirroring each other, showing the same thing, or turn off mirroring, which lets each display show its own image. That's how you want it set for QLab, so that you can see QLab on your display, and the audience sees your cues on the other display or displays. Amazingly, *there is no Terminal command for this!* To turn off display mirroring:

```
Open System Preferences;
Choose Displays;
Choose Arrangement;
Uncheck Mirror Displays.
```

Disable *Displays have separate Spaces*

Spaces is Apple's name for virtual desktops (if you don't know what this means, don't worry about it.) If your displays are set to have separate spaces, the Menu bar also appears on all Displays, and that is visible to your audience when no cues are playing through QLab. To set your displays to share Spaces, and thus keep the menu bar out of your picture, open a Terminal window and enter this command:

```
defaults write com.apple.spaces spans-displays -bool TRUE
```

Important: you'll need to log out, then back in again for this to take effect.

Blackout the Desktop

When QLab is playing a Video cue, it places a black "backdrop" over any screen that the Video cue is playing on. When no video is playing, however, QLab does not display this backdrop. Therefore, in order to prevent your audience from seeing anything when no Video cue is playing, you'll need to set the desktop background on your projector (or other audience-visible display) to black. You can do that in two ways. Either:

```
Open System Preferences;
Choose Desktop & Screen Saver;
Choose Desktop;
On your projector (or other display), choose "Solid Colors";
Click "Custom Color...";
Set the color to black.
```

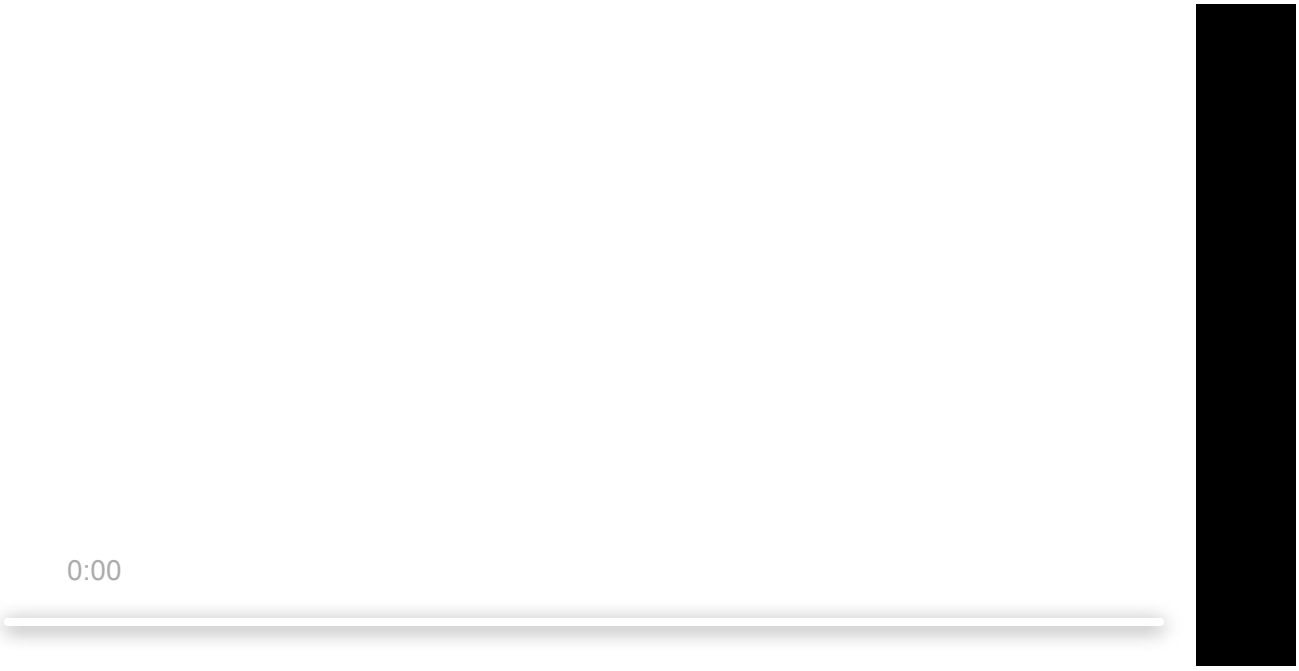
Alternately, QLab provides a quick and easy way to do the same thing. Simply choose *Black out desktop backgrounds* from the **Tools** menu, and all desktop backgrounds will be set to black. You can later choose *Restore saved desktop backgrounds*, also from the **Tools** menu, to restore the desktop backgrounds you had previously.

Getting Started

Welcome to the workspace

When you first open QLab, you're presented with a new, fresh document. QLab documents are referred to as **workspaces.** A workspace contains one or more **cue lists** which contain **cues.**

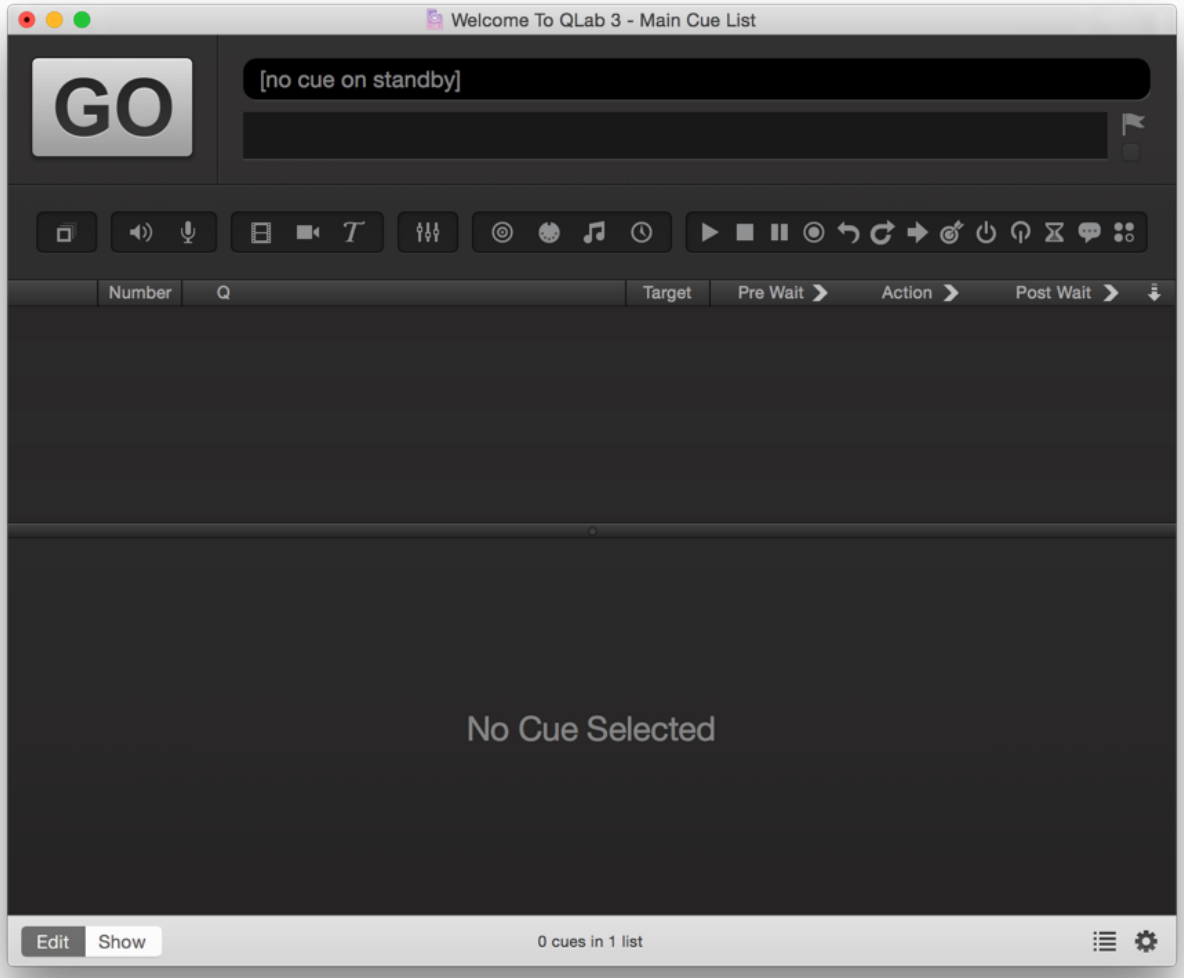
Detailed descriptions of the various cue types and cue lists can be found in other sections of the documentation; for now we will focus on what you see when you look at a workspace.



download video ↓

A note on style

- On this page only, every time a new tool, interface item, or concept that we feel is particularly essential is mentioned, it will appear in **bold text.** This is meant to help you notice that you're being introduced to a new idea. Thereafter, and throughout the rest of this documentation, bold text will be used in the traditional manner, as well as to indicate a menu name (such as the **File** menu.)

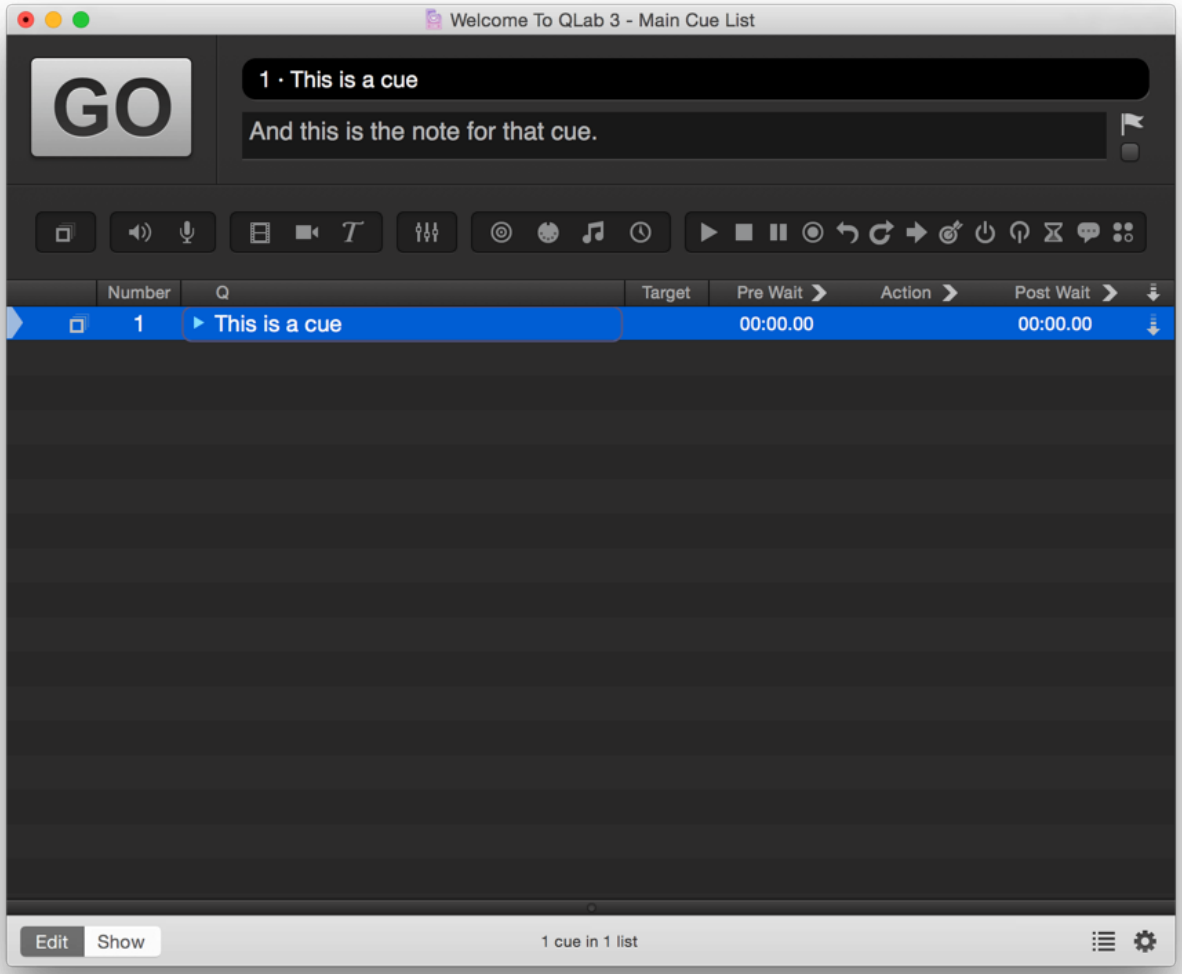


The GO Button

Prominently located in the top left corner of the workspace, the **GO button** starts, or **triggers**, the cue which is currently **standing by** at the **playback position.** The playback position then advances to the next cue, which will then be standing by, displayed in the **standby indicator** and ready to GO. The default keyboard shortcut for the GO button is the **space bar**. You can change this shortcut in Settings.

The Standby Indicator

Located at the very top of the workspace across most of the width of the window, the standby indicator displays the **cue number** and **cue name** of the cue in the playback position. In other words, it tells you what cue will play the next time the cue list is triggered. When a cue is in standby, it will also appear highlighted in the cue list, with a small indicator arrow against the left edge of the workspace.



Beneath the standby indicator is the **Notes field** which displays notes associated with the currently standing-by cue.

The Notes Field

Beneath the standby indicator and to the right of the GO button is the **Notes field.** Text entered in this field is connected to the currently standing by cue, and is visible whenever that cue is standing by, so it is the perfect place for notes or

special instructions to your operator. Text in the Notes field is searchable using the **find** feature.

Flagged Cues

Next to the Notes field is the **Flagged checkbox**. Flagging a cue is a way of marking it for later. You might, for example, flag cues during a run through of your show as a way of notating which ones need to be reviewed after the run through.

The Toolbar

The **toolbar,** found underneath the Notes field, is a ribbon of icons for each of the different cue types available in QLab. Clicking any of these icons will create a new cue of that type. Explanations of the different cue types can be found later in this documentation.

The Cue List

Beneath the toolbar is the **cue list.** It is composed of eight columns that provide information about each cue.

Playback Position

Along the left edge of the cue list is the playback position indicator (a right-facing grey pointer) which shows you which cue is on standby.

Cue Status

The left-most column displays the status of the cue:

- A green triangle means the cue is **active.**
- A yellow circle means the cue is loaded and is ready to be triggered.
- A red X means the cue is broken and cannot be played.
- A grey triangle outline means the cue has been stopped, but has an effect that must finish rendering before the cue can be started again.

This column also displays an icon depicting the cue's type; these icons match the cue icons in the toolbar.

Cue Number

A cue number may be any text string, or may be empty. All cue numbers in a given workspace must be unique. Cue numbers do not need to be consecutive, nor do they need to be digits. Acceptable cue numbers could be "1," "1.5," "A," "AA," "A.5," "Preshow Music," or "Fred." Change the number of a cue by double clicking in the cue's number column, or by selecting the cue and using the keyboard shortcut **N**.

It's important to realize that since cue numbers are text strings, "1", "1.0", and "1.00" count as three different unique cue numbers in QLab.

Renumbering Selected Cues

Select *Renumber Selected Cues* in the **Tools** menu, or use the keyboard shortcut ⌘**R**, to automatically assign new numbers to each currently selected cue in the workspace. Note that since cue numbers must be unique within the workspace, the renumber tool will skip numbers that already exist.

Cue Name

A cue name may be any text string, or may be empty. The name of a cue will default to reflect the name of its **target**; for Audio and Video cues, that is the name of the target file. For Fade cues, it will be the word “fade” plus the name of the cue that the Fade targets. Other types of cues have their own, hopefully logical, default names. You can change the name of a cue by double clicking in the cue’s name column, or by selecting the cue and using the keyboard shortcut **Q**. Unlike cue numbers, cue names do not have to be unique.

Note that changing the name of a cue will also change the name of any cues that target it (for example, Fade cues), if those cues are using their default names.

Target

A key concept in QLab is that some types of cues have a **target** which is the recipient of the action of that cue. So, Audio and Video cues have targets, which are media files. When an Audio cue is triggered, the target file is played. Fade cues have targets, which are other cues which have fade-able parameters, such as audio levels or video display geometry. When a Fade cue is triggered, the parameters of the target cue are faded. For cues that require a target, they must have one and only one target.

Thus, the target column shows different information for different types of cues.

For Audio and Video cues, the Target column displays a round button that looks like an upwards-pointing arrow inside a circle. Clicking this arrow will open a Finder window in which you can select the file you wish to target. You can also set the target of an Audio or Video cue by dragging and dropping an appropriate file onto the cue from the Finder.

Cues which target other cues, such as Fade cues and Stop cues, will display the cue number of their target cue. If the target cue has no number, the cue name will be displayed instead. If the cue lacks a target, the target column will display a question mark. You can assign a target to these sorts of cues by typing a cue

number into the Target column, by dragging and dropping the cue onto its intended target, or by dragging and dropping the intended target cue onto the cue.

The default keyboard shortcut for changing the selected cue's target is **T**.

Cues which do not require a target will show nothing in this column.

Pre-wait

Pre-wait is the amount of time that QLab waits between receiving a trigger for a cue and starting the action of that cue. For example, an Audio cue with a pre-wait of 3 would start playing sound three seconds after being triggered.

The pre-wait of a cue can be edited by double clicking and typing in the pre-wait column, or by using the keyboard shortcut **E**.

Action

The **action** of a cue tells you how long it takes for the cue to complete, not counting pre-wait. Action is often used interchangeably with "duration" conversationally. They are one and the same. The action of some cues cannot be edited directly, but for those that can, they can be edited by double clicking and typing in the action column, or by using the keyboard shortcut **D**.

Post-wait

The **post-wait** of a cue is meaningful only in combination with an as-yet-un-discussed feature: **auto-continue.** When a cue is set to auto-continue, then it triggers the next cue in the cue list when it is itself triggered. If the first cue has a post-wait time, then, once the first cue is triggered, QLab waits for the post-wait to elapse and then triggers the second cue.

The post-wait of a cue can be edited by double clicking and typing in the post-wait column, or by using the keyboard shortcut **W**.

Notes about pre-wait, action, and post-wait.

The pre-wait, action, and post-wait columns display seconds with two decimal places, but QLab is accurate to three decimal places.

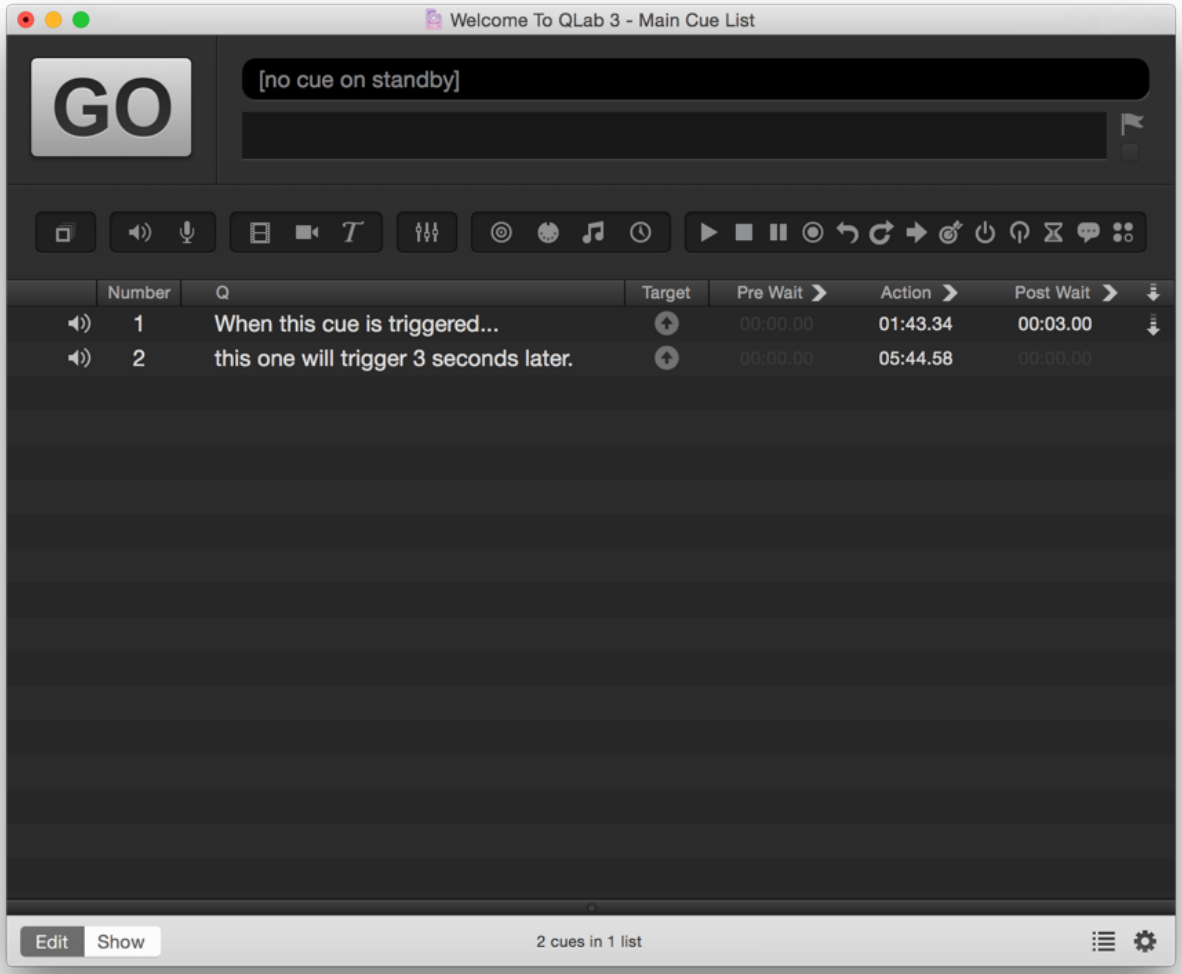
When a cue is not playing, these columns will always display the total duration of the cue or its wait. When a cue is playing, you can toggle between seeing how far into a cue (or wait) you are and how far from the end of a cue (or wait) you are by clicking the arrows in the column headers.

It's also important to remember that in the interest of keeping QLab's interface from using too much processing power, QLab sometimes updates times as little as once per second, which means that the times displayed may appear to be wrong by as much as one second. This does not indicate that the actual behavior of QLab is off by a second, only the display. A paused cue will always display exact times.

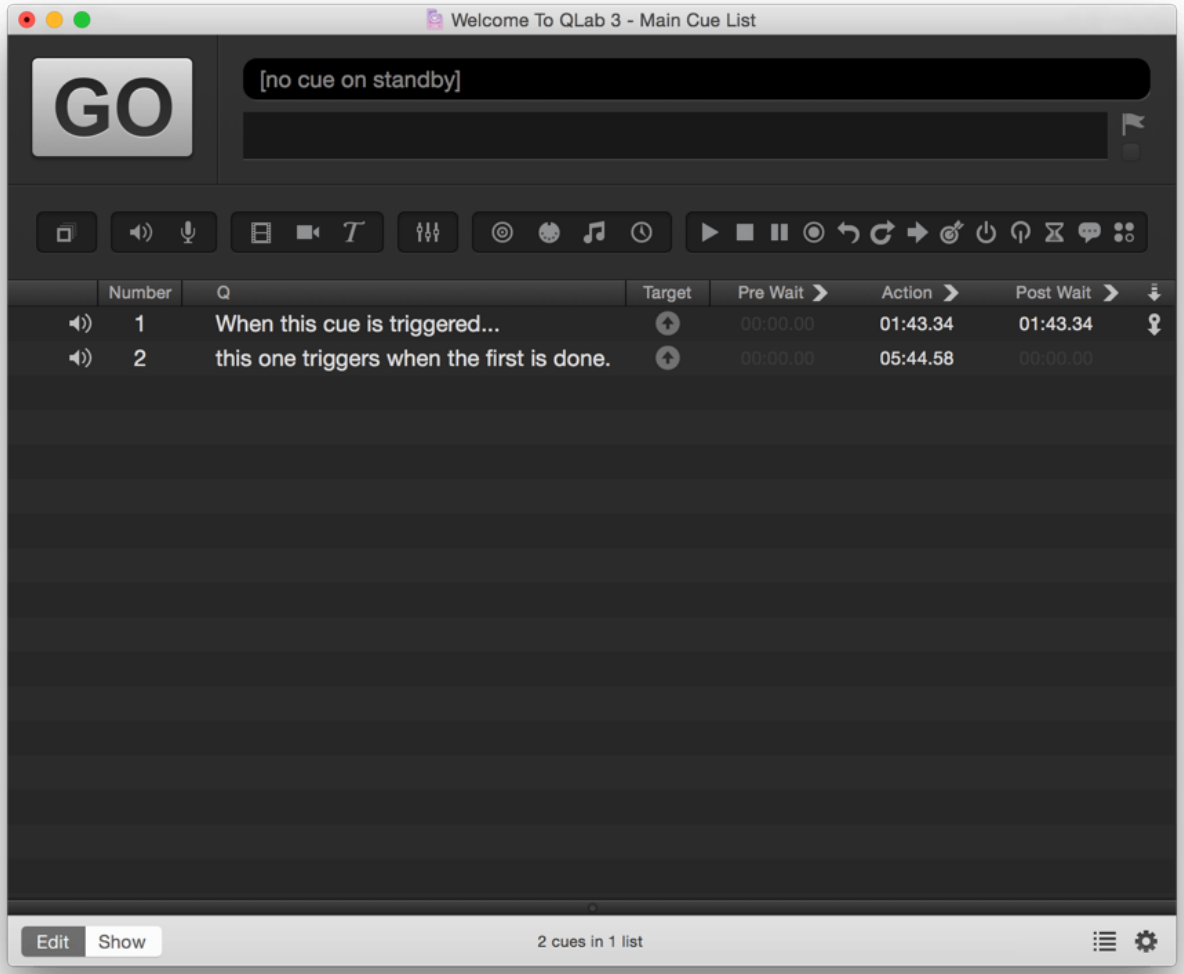
Auto-follow and Auto-continue

The final column, labeled with a downwards-pointing arrow in the column header, displays icons indicating whether a cue has been set to **auto-continue** or **auto-follow.**

If a cue is set to auto-continue, as soon as the cue is triggered the next cue in the cue list will trigger as well. If the cue has a post-wait as well as an auto-continue, the post-wait will be honored before the next cue is triggered.

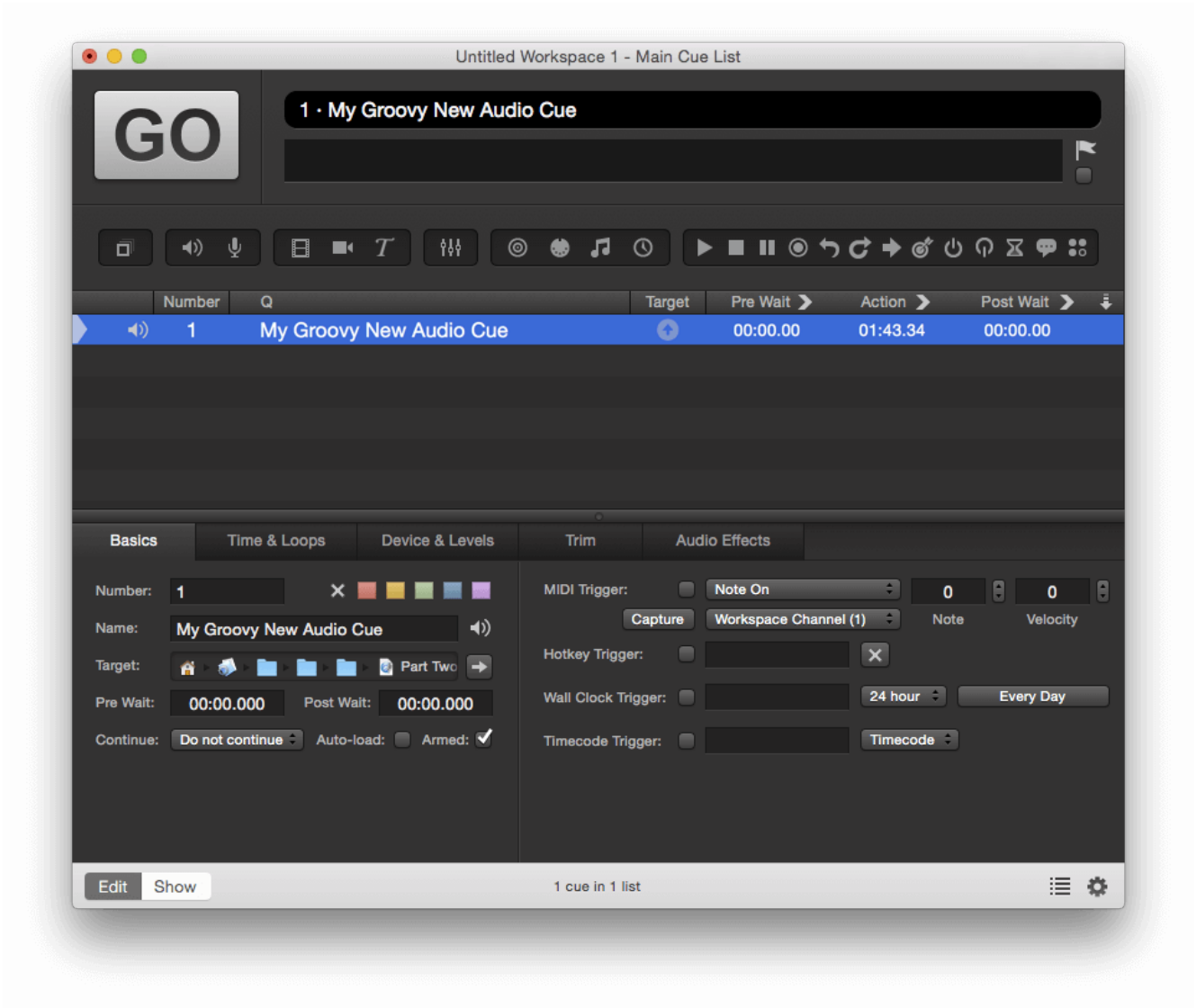


If a cue is set to auto-follow, then the next cue will be triggered as soon as the first cue completes. When you set a cue to auto-follow, QLab will automatically show a post-wait time equal to the action of the cue. This cannot be edited, and serves as a visual reminder of the auto-follow.



The Inspector

The **inspector** is located beneath the cue list, and can be shown or hidden by selecting *Inspector* from the **View** menu or using the keyboard shortcut ⌘**I**. The inspector is a tabbed interface. The **Basics tab** is the same for all cues and is a convenient place to adjust cue parameters such as number, name, and target. Other tabs vary based on what type of cue is selected, and will be explained later in this documentation.



The settings on the left side of the Basics tab refer to basic properties of the cue. Many of the properties shown in the Basics tab are also editable from the Cue list.

- **Number.** Discussed above.
- **Color.** The cue can be given a background color to make it stand out in the cue list. There are five colors to choose from; the X sets the background to no color. The color doesn't alter the cue in any way, so feel free to adapt color coding in whatever way is most useful to you.
- **Name.** Discussed above.
- **Target.** This field, only relevant when the selected cue has a target, shows either the name of the target cue, or the full path to the target file depending on what type of cue is selected. Clicking the arrow button next to the target field will show you the target, either by jumping the cue list to the target cue, or opening a Finder window to show the target file.
- **Pre Wait.** Discussed above.
- **Post Wait.** Discussed above.
- **Continue.** Discussed above.
- **Auto-load.** If this box is checked, this cue will automatically be loaded after the previous cue is played. A global setting is available to set all new cues to auto-load by default.
- **Armed.** Cues are armed by default. When disarmed, cues will not execute their action, but an auto-continue or auto-follow will be triggered if present. Disarming a cue is useful, for example, for temporarily silencing a cue while allowing the surrounding cues to operate as-is.

download video ↓

The settings on the right side of the tab refer to four ways to trigger a cue directly. You can use these triggers in any combination.

- **MIDI Trigger.** Individual cues can be triggered with external hardware or software using MIDI voice messages here. Check the box to enable the trigger, and then either program in the message manually, or click the *Capture* button to record an incoming message. MIDI trigger values can include > and < operators, or be set to "any". This is particularly helpful for MIDI Note On messages, where a specific note velocity won't necessarily be achieved with each press of the MIDI controller. Set the velocity (byte 2) to "any", and the note will be the specific part of the message, rather than being limited to a particular velocity along with the note. Then, no matter how hard you press the piano key (as an example), the cue will start.
- **Hotkey Trigger.** The spacebar is the default keyboard command to start the cue in the playback position (the GO button), but most keys on the keyboard can become a trigger for a given cue regardless of its playback position. Note that some keys are reserved by QLab, such as **Esc** (panic/stop all), and keys that have already been assigned functions in Key Map Settings. Hotkey triggers behave just like regular keyboard shortcuts for menu items; for example, by assigning the Hotkey "J" to cue 49, any time you press **J** on the keyboard, cue 49 will start even if it is not currently on standby. Hotkeys may include modifier keys (option, control, function, and shift). Hotkeys are particularly useful for triggering scripts that act on selected cues. Click the *X* to clear the hotkey assignment.
- **Wall Clock Trigger.** This will trigger a cue at a specific time of day and, optionally, only on certain days of the week. Please note that computer clocks will drift if left to their own devices, so if your show computer is offline or if you've disabled online clock setting, be sure to check your clock's accuracy from time to time.
- **Timecode Trigger.** Cues can be triggered from incoming LTC or MTC timecode, but only if timecode has been enabled for the enclosing cue list. You can enable timecode for a Cue List in the **Sync tab** of the inspector when the cue list is selected. The dropdown menu lets you choose to enter

the timecode trigger using either the timecode format of hours:minutes:seconds:frames or the real-time format of hours:minutes:seconds:decimals.

download video ↓

Workspace Footer

Edit Mode and Show Mode

On the left side of the workspace window footer, two buttons allow you to toggle between **Edit mode** and **Show mode**. When a workspace is in show mode, the following functions of QLab are disabled:

- The inspector
- The toolbar
- Load-to-time
- Find
- Adding, deleting, or reordering cues
- Editing any cue properties (name, number, target, times, etc.)
- Copy and paste levels
- Copy and paste geometry
- Copy and paste fade shape

Just as important is the list of things which are *not* disabled in show mode:

- The Audition window
- Opening, closing, and saving workspaces
- Use of the **ESC** key
- Showing or hiding the Cue lists & active cues display
- Viewing and changing workspace settings

Show mode is a safety mechanism designed to prevent accidental changes to a workspace, not a security mechanism to prevent deliberate changes.

Cue and Cue List Count

The center of the footer displays the number of cues in the workspace and the number of cue lists into which they are divided.

Broken Cues and Warnings

The Broken Cues icon will appear in the bottom right corner of the workspace window when (and only when) the workspace contains broken cues. Clicking this icon displays a list of all broken cues, explains why they're broken and provides instructions for how they can be fixed. Learn more about this in the Workflow Tools section.

Sidebar

Click the Sidebar button in the bottom right corner of the workspace window to show or hide the sidebar. Learn more about this in the Workflow Tools section.

Settings

The gear in the bottom right corner of the workspace allows you to access its settings. Learn more about this in the Settings section.

Making cues

There are several ways to make a cue in QLab:

1. Click on one of the cue type icons in the toolbar.
2. Drag a cue type icon from the toolbar into the cue list.
3. Double-click on a cue type in the toolbox.
4. Drag a cue type from the toolbar into the cue list.
5. Select a cue type from the **Cues** menu.
6. Use one of the **⌘-number** keyboard shortcuts for the first ten cue types.

There are also ways to make cues using OSC and AppleScript, which you can learn more about in the OSC Dictionary and AppleScript Dictionary sections of this documentation.

Targeting Audio and Video cues

Audio and Video cues require a target which is a file containing the relevant type of media. There are several ways to assign a target to an Audio or Video cue:

1. Click the *target button* for the cue in the cue list.
2. Select the cue and type the target hotkey, *T* by default.
3. Double-click in the target field in the *Basics* tab of the inspector.
4. Drag and drop a file from the Finder onto the cue in the cue list.
5. Drag and drop a file from the Finder into the target field in the Basics tab of the inspector.

Targeting other cues

Cues which target another cue include Fade, Start, Stop, Pause, Load, Reset, Devamp, GoTo, Target, Arm and Disarm. There are several ways to assign a target to these cues:

1. Double-click in the target column in the cue list.
2. Select the cue and type the target hotkey, *T* by default.
3. Double-click in the target field in the *Basics* tab of the inspector.
4. Drag and drop the intended target cue onto the cue in the cue list.

Rules for cues

Once started, cues continue to play until they reach the end of their programmed action, or until they're told to stop. For an Audio or Video cue, a cue's action is the duration of the media file which it targets unless you program it otherwise. For a Fade cue, the default duration is five seconds.

When you start a cue, the playback position will advance to the next cue. On the next press of the GO button or space bar, the next cue will start. If you're accustomed to using a non-theatrical playback program such as iTunes, this behavior can be disorienting at first. Fear not, for this is how QLab is meant to work.

Rules for control

QLab can be told to GO, which is to say, to trigger the cue or cue sequence that is currently standing by, and advance the playback position to the next cue or cue sequence, in a number of ways. You can:

1. Click on the GO button with the mouse.
2. Press `space` (or whichever keyboard shortcut you assigned to GO.)
3. Send an OSC "/go" command to QLab from another program or an external device.
4. Send another OSC command that you assigned to "GO" in OSC Controls.
5. Send an MSC "Go" command to QLab from another program or an external device.
6. Send a MIDI message that you assigned to "GO" in MIDI Controls.
7. Run an AppleScript which sends the "go" command to QLab.

There are similar options for other commands such as stop, panic, pause, and so forth. The purpose of this plurality is to accommodate the myriad situations and possibilities that QLab might encounter in the world, and make it easier for you to use QLab in the way that suits your needs or your style.

Keyboard Shortcuts

Key Symbols

Symbol	Meaning
⌘	command
⇧	shift
^	control
⌥	option
⌫	delete

Default Key Map

Command	Key
GO	space
Panic All	escape
Hard Stop All	esc esc
Pause All	[
Resume All	]
Preview	V
Load Selected Cue	L
Pause/Resume Selected Cues	P
Panic Selected Cues	S

Command	Key
Hard Stop Selected Cues	SS
Edit Cue Number	N
Edit Cue Name	Q
Edit Cue Target	T
Edit Cue Pre-wait	E
Edit Cue Action (duration)	D
Edit Cue Post-wait	W
Cycle Cue Continue Mode	C
Flag/Unflag	F

Menu Keyboard Shortcuts

QLab

Command	Key
Hide QLab	⌘H
Hide Others	⌥⌘H
Quit QLab	⌘Q

File

Command	Key
New Workspace	⌘N
New From Template	⇧⌘N
Open Workspace...	⌘O
Close	⌘W

Command	Key
Save	⌘S
Save As...	⇧⌘S

Edit

Command	Key
Undo	⌘Z
Redo	⇧⌘Z
Cut	⌘X
Copy	⌘C
Paste	⌘V
Delete	⌘⌫
Select All	⌘A
Find...	⌘F
Find Next	⌘G
Find Previous	⇧⌘G
Show Fonts	⌥⇧⌘T
Bigger	⌘+
Smaller	⌘-
Show Colors	⌥⇧⌘C
Copy Style	⌥⌘C
Paste Style	⌥⌘V

Cues

Note: You can reassign keyboard shortcuts for cues by re-arranging them in the toolbox.

Command	Key
Group	⌘0
Audio	⌘1
Mic	⌘2
Video	⌘3
Camera	⌘4
Titles	⌘5
Fade	⌘6
OSC	⌘7
MIDI	⌘8
MIDI File	⌘9

Tools

Command	Key
Load to time...	⌘T
Renumber selected cues...	⌘R
Delete numbers of selected cues...	⌘D
Jump to cue...	⌘J
Jump to selected cue's target	⇧⌘J
Toggle live fade preview	⇧⌘P

Some tools are only visible when certain cue types are selected:

Command	Key	Cue Type(s)
Copy Levels	⇧⌘C	Audio, Mic, Fade
Paste Levels	⇧⌘V	Audio, Mic, Fade
Copy Integrated Fade Shape	^⌘C	Audio, Video, Camera, Titles
Paste Integrated Fade Shape	^⌘V	Audio, Video, Camera, Titles
Copy Geometry	⇧⌘C	Video, Camera, Titles
Paste Geometry	⇧⌘V	Video, Camera, Titles
Set Levels From Target	⇧⌘T	Fade
Copy Fade Shape	^⌘C	Fade
Paste Fade Shape	^⌘V	Fade
Revert Fade Action	⇧⌘R	Fade

View

Command	Key
Toggle Full Screen	⇧⌘F
Inspector	⌘I
Toolbox	⌘K
Cue Lists & Active Cues	⌘L
Toggle Between Cue Lists & Active Cues	⇧⌘L
Broken Cues & Warnings	⌘B
Workspace Settings	⌘,
Select Next Cue	⌘↓
Select Previous Cue	⌘↑
Move Playback Position To Next Cue	⇧⌘↓
Move Playback Position To Previous Cue	⇧⌘↑

Command	Key
Select Next Inspector Tab	⌘→
Select Previous Inspector Tab	⌘←
Enter Edit Mode	⇧⌘[
Enter Show Mode	⇧⌘]

Window

Command	Key
Minimize	⌘M
Audition Window	⇧⌘A

Other keyboard shortcuts

Command	Key
Move Playback Position To Next Cue Sequence	+
Move Playback Position To Previous Cue Sequence	-
Expand all group cues	>
Collapse all group cues	<

Workflow Tools

Broken Cues & Warnings

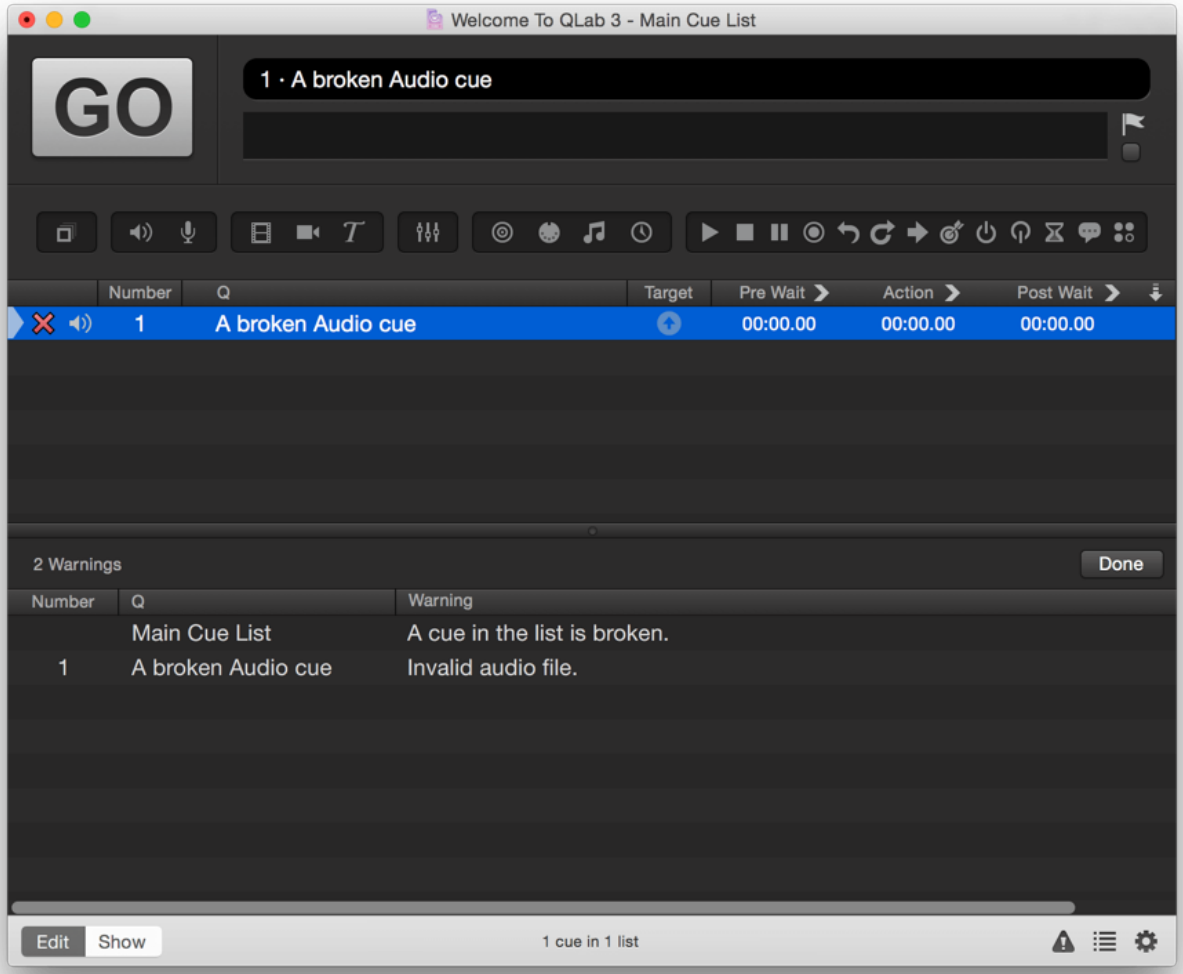
download video ↓

If a cue is missing one or more pieces of information it requires for playback, it will be flagged as a broken cue. Broken cues have a red X in the leftmost column of the cue list, and hovering your mouse over that red X shows a tooltip which gives a brief description of the problem.

Here are some common examples of problems:

- A target has not been assigned to a cue which requires one.
- A Fade cue has no activated parameters to fade.
- A Group cue contains one or more broken cues within it.
- An Audio cue has not been assigned to an audio patch.
- There is a problem with the audio patch to which an audio cue has been assigned.
- A Video cue has not been assigned to a surface.
- There is a problem with the surface to which a Video cue has been assigned.
- A MIDI, MIDI file, or Timecode cue has not been patched to a destination.
- There is a problem with the destination to which a MIDI, MIDI file, or Timecode cue has been assigned.
- The cue requires a license which is not installed or activated.

If your workspace contains any broken cues, a triangular indicator with an exclamation point appears on the right side of the footer (in addition to the red X beside each broken cue). Clicking the indicator opens the “Broken Cues and Warnings” panel, which contains a list of all the broken cues and a brief description of why they’re broken. Clicking on a cue in that list will jump the selection in the cue list to that cue.



A broken cue will behave like a disarmed cue when it comes to being played as part of a cue sequence. Its pre-wait, post-wait, and follow state will be respected wherever possible.

Cues which are flagged also appear in the broken cues and warnings panel. Flagged cues aren't necessarily broken, they're just listed for the purpose of having ready access to all flagged cues.

When a broken cue is fixed, its red X will disappear. When a workspace contains no broken cues and no flagged cues, the indicator in the footer with disappear as well.

The Cue Lists & Active Cues Sidebar

The Cue Lists & Active Cues sidebar, hidden by default, can be shown by choosing *Cue Lists & Active Cues* from the **View** menu, or by using the keyboard shortcut ⌘**L**. The sidebar shows four transport control buttons at the top, and two lists below them. You can click the headings of these two lists to switch between them, choose *Toggle Between Cue Lists & Active Cues* from the **View** menu, or use the keyboard shortcut ⇧⌘**L**.

(The inspector, shown in most screenshots in this documentation, is hidden here

for clarity.)

The Buttons

- **Stop All.** Stops all currently playing cues immediately, and unloads any loaded cues.
- **Reset All.** Returns the playback position to the top of the currently active cue list, and resets any temporarily changed properties of cues to their default values.
- **Pause All.** Pauses all currently playing cues.
- **Resume All.** Resumes all currently paused cues.

Cue Lists

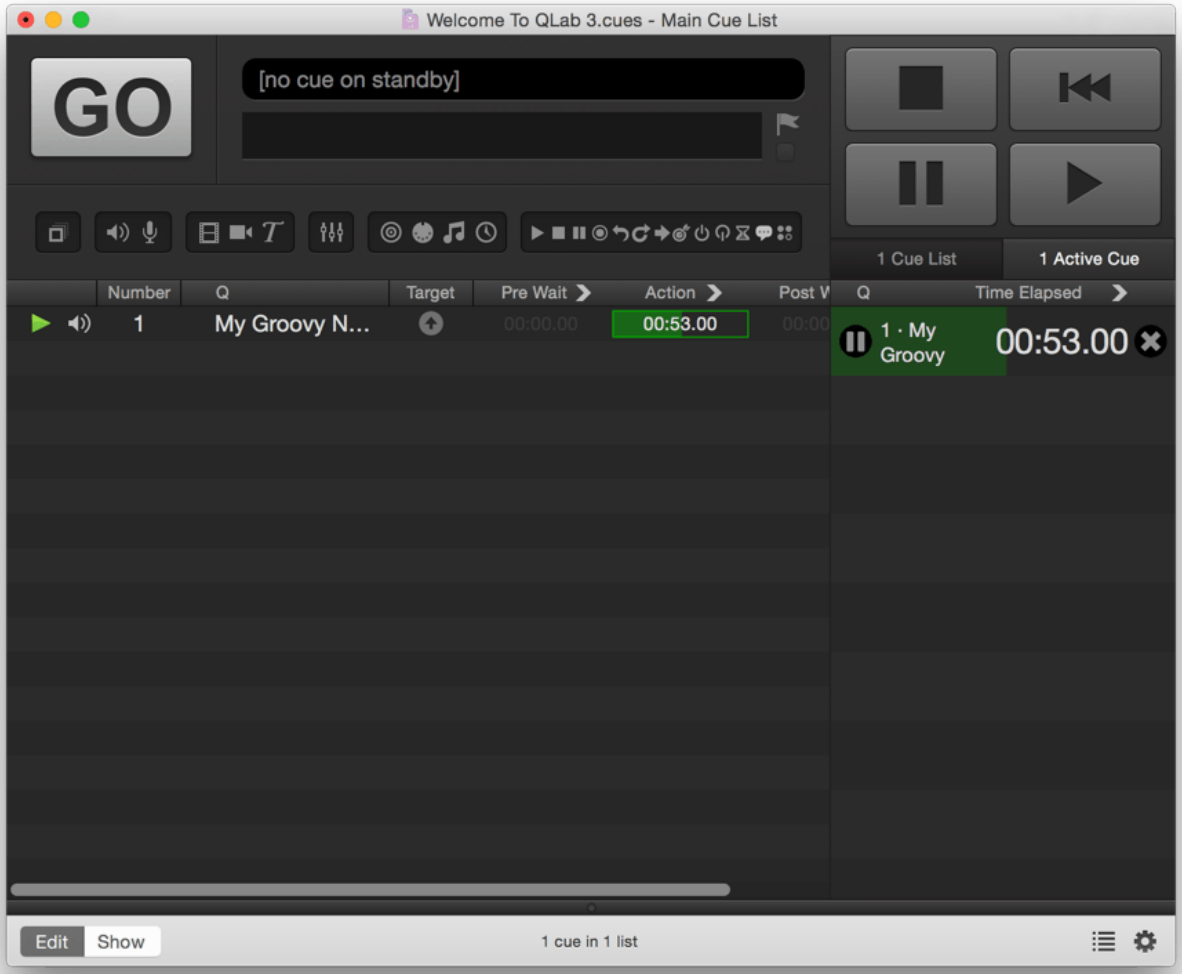
A new workspace will have one cue list named “Main Cue List.” You can change this name as needed. To add a new cue list, click the button at the bottom of the sidebar labeled *+ Add New List*. To delete a cue list (and all the cues in it!), select the cue list you want to delete and click the *Delete* button, which relabels itself to match the name of the selected list.

When a cue list is selected, the cues within that list are displayed in the main cue list view, and that cue list is considered to be the **active cue list.**

You can also click the disclosure triangle next to the name of a cue list to show or hide the cues within that list in the sidebar.

Active Cues

The Active Cues list shows all cues which are currently playing or paused.



Each active cue is displayed in its own row, with its own controls. The row shows the cue number and name, and the elapsed or remaining time in larger text. You can toggle between showing elapsed time and remaining time by clicking on the chevron in the list header.

On the left side of each row is a pause/resume button, on the right side is a panic button, and throughout the whole row is a progress bar, colored green when the cue is playing and yellow when the cue is paused, which gives a visual indication of the progress of the cue.

If you hover your mouse over an active cue, the leading edge of the progress bar will show a thick yellow line, which can be dragged left or right to scrub the cue backward or forward.

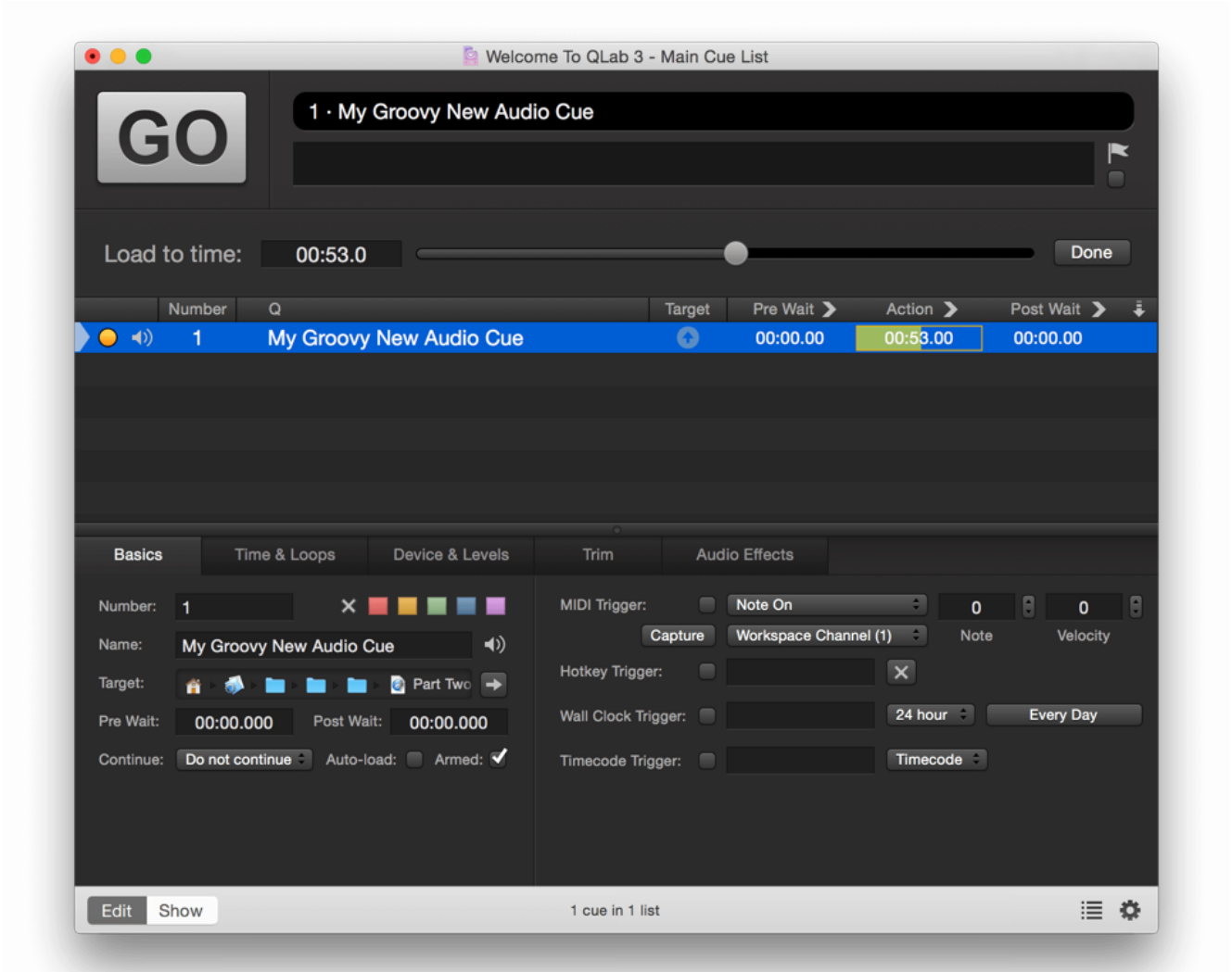
Load To Time

download video ↓

The Load to Time tool allows you to “fast forward” through the standing-by cue or cue sequence so that it is loaded to particular time, and will begin playing from that time when the GO button is pressed.

To access this feature, select *Load to Time* from the **Tools** menu, or use the keyboard shortcut ⌘**T**. *Load to Time* is one of the tools that occupies toolbar, and the menu or keyboard shortcut cycles through three states:

1. When the Load to Time tool isn't displayed, the command will display it.
2. When the Load to Time tool is displayed but the cursor isn't in its text box, the command will move it there.
3. When the Load to Time tool is displayed and the cursor is in the text box, the command will close the Load to Time tool and move focus back to the cue list.



To use the Load to Time tool, type a number into the text field or drag the slider along the timeline to load the selected cue or cue sequence to your desired time.

If you type a negative value into the field, QLab will load to that amount of time back from the end of the cue or sequence, if possible.

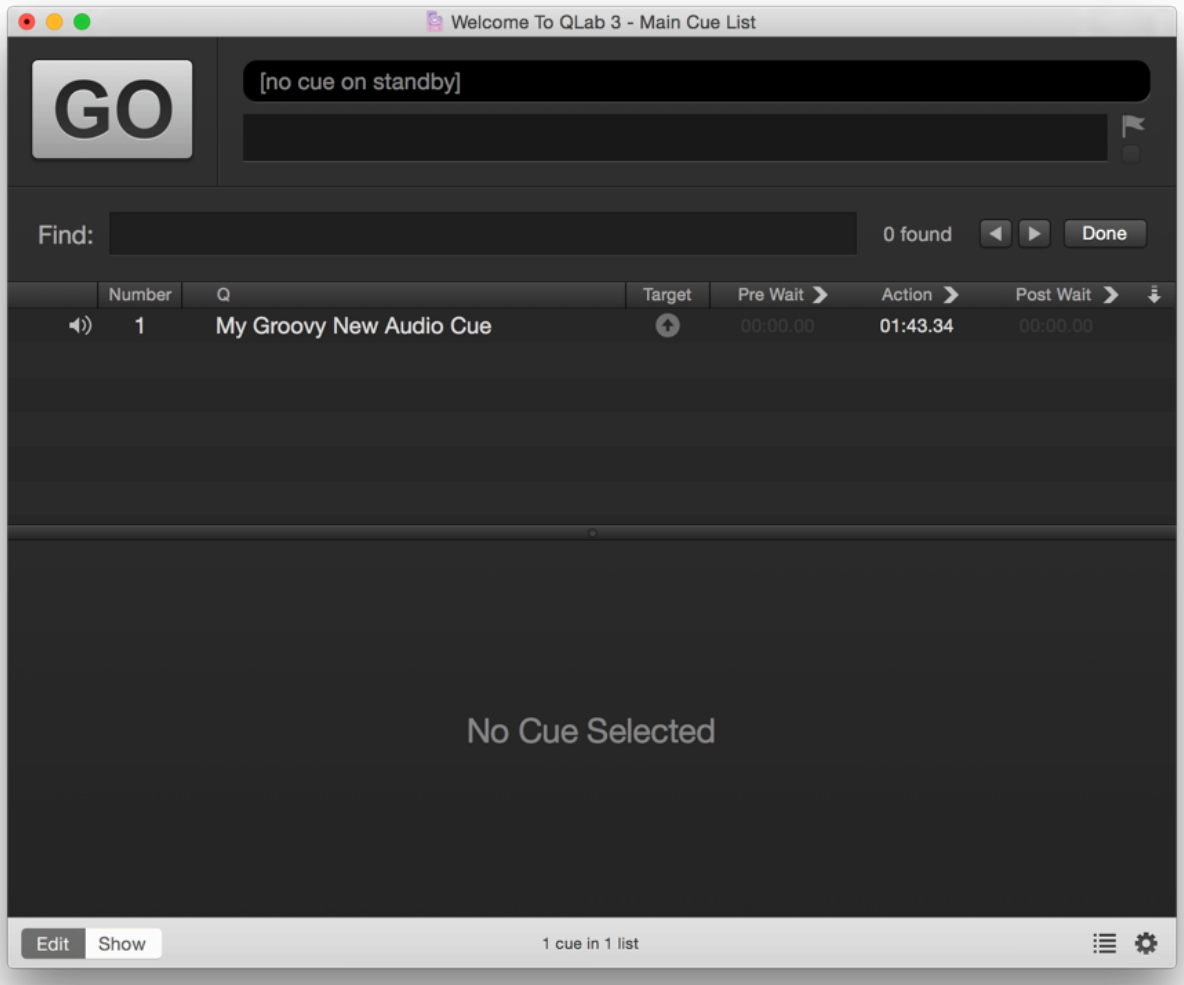
Find

download video ↓

To search the workspace, select *Find* from the **Edit** menu in the toolbar, or use the keyboard shortcut is ⌘**F**. This feature searches cue names, cue numbers, the file names of audio and video cue targets, and any notes you've added (in the current cue list only).

Find is one of the tools that occupies toolbar, and the menu or keyboard shortcut cycles through three states:

1. When the Find tool isn't displayed, the command will display it.
2. When the Find tool is displayed but the cursor isn't in its text box, the command will move it there.
3. When the find tool is displayed and the cursor is in the text box, the command will close the Find tool and move focus back to the cue list.

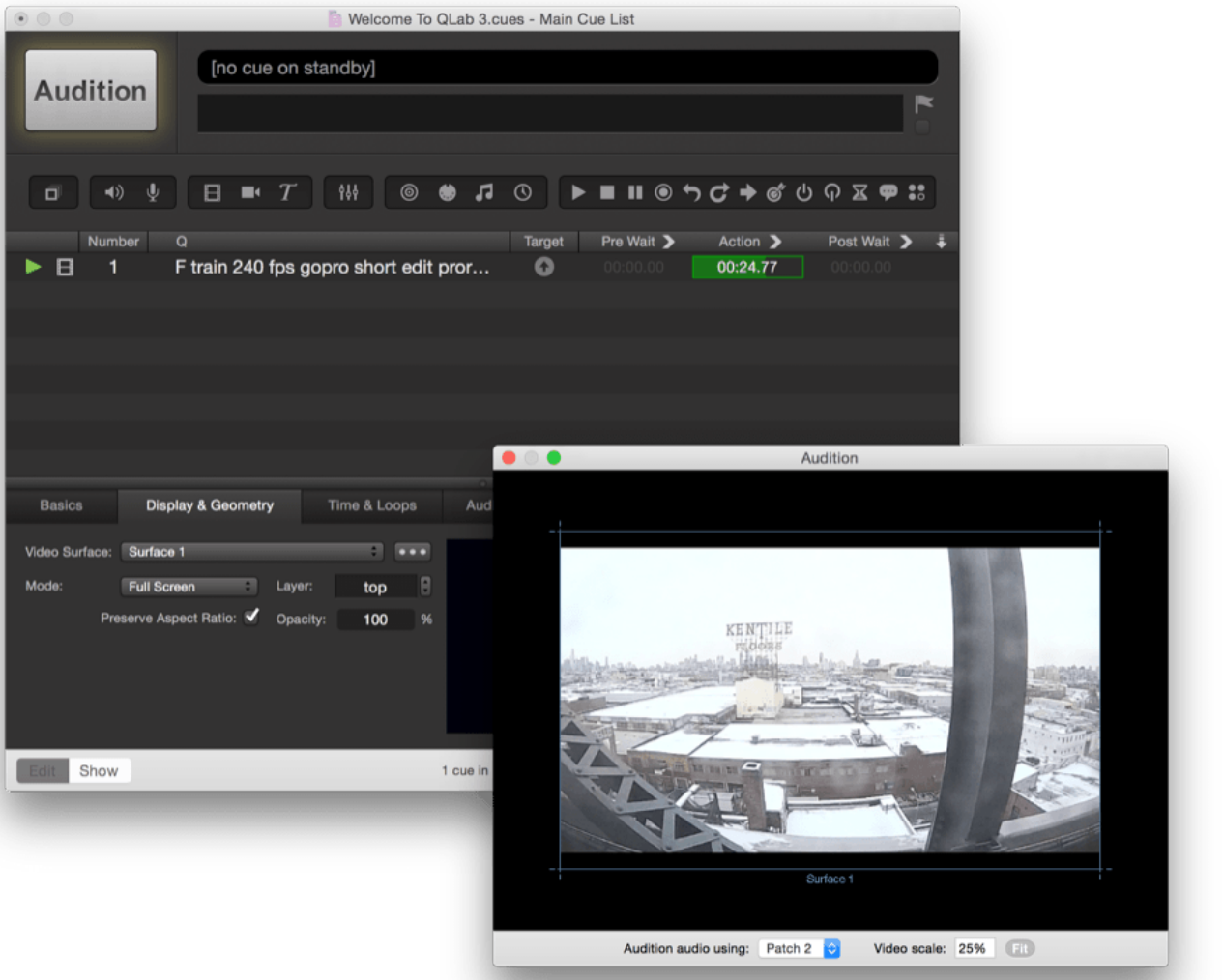


The Audition Window

download video ↓

The Audition Window is a tool to listen to or watch cues privately without running them through your entire sound or video system. To access the audition window, select *Audition Window* from the **Window** menu or use the keyboard shortcut ⇧⌘**A**.

When the Audition Window is open, the GO button is replaced with a glowing “Audition” button to indicate that triggering new cues will audition them rather than playing them normally.



The Audition Window has its own audio patch selector; any cues started when the audition window is open will play their audio to that patch instead of the patch selected in the cue.

The Audition Window audio patch is a convenient way to simulate a headphone or pre-fader level (PFL) bus: if you're using patch 1 for your audio interface, then you can assign patch 2 to your headphone jack or other local monitoring output, and hear the audio out of that alternate device by opening the Audition Window.

While the Audition Window is open, Video cues appear in the window instead of on their regularly assigned surface. If Video cues played to multiple surfaces are

played at the same time, they are shown side by side in the Audition Window.

Any cues already playing when the Audition Window is opened will continue on their normal outputs. Only newly started cues are sent to the Audition Window and its audio patch.

You can scale your video output by any desired percentage using the scale control in the footer of the Audition Window, or click *Fit* to have QLab set the scale according to the current size of the window. It's best to click *Fit* while a Video cue is playing, otherwise QLab doesn't know what to measure against.

To exit the Audition Window, simply close it or type ⇧⌘**A** again, and your workspace will return to normal. Any cues which were started while the Audition Window was open will stop, and any cues which were started before the Audition Window was opened will remain unchanged.

Group Cues

Group cues hold a unique position in QLab, because they contain other cues. In this way, they are quite like cue lists, and in fact cue lists are indeed a special kind of Group cue.

The default keyboard shortcut to create a group is ⌘**0**. If you create a Group cue while one or more other cues are selected, those cues will be placed inside the Group. Once a group is created, it can be collapsed or expanded for visual simplicity, using the gray disclosure arrow in the upper-left corner.

The inspector shows two tabs for group cues: Basics and Mode.

Basics

Please refer to the section on the inspector in the **Getting Started** section of this documentation.

Mode

A group cue can have one of four modes:

Start first child and enter into group.

Cues grouped in this mode will appear in a blue outline with round corners. Cues within this type of group won't automatically affect each other or trigger each

other. When the group is triggered, the first cue within it will play and the playback position advances to the next cue in the list. This is essentially an organizational tool to visually separate cues into different sections within the cue list, and to hide or show an entire group of cues with a single click (on the grey disclosure arrow in the upper-left corner of the box).

Start first child and go to next cue.

Cues grouped in this mode will appear in a blue outline with square corners. Cues within this type of group will get skipped over unless they are connected with auto-continues or auto-follows. Before auto-continues or auto-follows are added, this is equivalent to disarming all but the first cue in the group, except it now includes a way to collapse the whole group at once. Once auto-continues and auto-follows are added, it's a tool to begin a timeline-like sequence of cues with only one press of the GO button.

Start all children simultaneously.

Cues grouped in this mode will appear in a green outline with square corners. Cues within this type of group will all start at the same time when GO is pressed. Using a start-all group in conjunction with pre-waits is another way to create a timeline-line sequence. For a more detailed description of how this can be achieved, please see the "Building cue sequences" section of the documentation.

Start random child and go to next cue.

Cues grouped in this mode will appear in a purple outline with square corners. When a group in this mode is triggered, it will randomly select any cue within the

group that is both armed and not currently playing and start that cue. **Note:** in versions prior to QLab 3.1, this mode would start any child regardless of its status as already active or disarmed.

Broken Cues

Group cues can become broken for the following reasons:

A cue in the group is broken.

Fix the cue inside the Group, and the Group will be fixed as well.

Timecode triggers require a Pro license.

You'll need to either install a Pro Audio or Pro Bundle license, or remove the timecode trigger from this cue.

Cue Sequences

A key concept in QLab is the cue sequence. A cue sequence is a series of cues that are triggered together from one single press of the GO button (or one single incoming MIDI command, MSC command, OSC command, hotkey press, etc.) Cue sequences can be built in three different ways:

- By connecting cues with auto-follows
- By connecting cues with auto-continues
- By putting cues into a start-all group

Auto-follow

An auto-follow happens after the duration of the first cue. For example, say cue 10 has a duration of three seconds and an auto-follow to cue 11. When you press GO, QLab will trigger cue 10, wait three seconds, and then trigger cue 11 automatically. If you change the target of cue 10 to a file that is longer or shorter, or adjust the duration of cue 10, QLab will adjust the auto-follow time on its own.

To create an auto-follow, select *Auto-follow* from the drop-down menu in the bottom-left corner of the Basics tab in the inspector. An arrow with a circle on top will appear in the far-right column of that cue's row in the cue list. To delete the auto-follow, select *Do not continue* from the drop-down menu.

Instead of using the drop-down menu, you can also click in the rightmost column of the cue list, where the icon appears, to cycle through auto-follow, auto-

continue, and do not continue, and the drop-down menu will adjust itself accordingly.

You can also use the keyboard shortcut for setting the selected cue's continue mode, which by default is **C**.

Auto-continue

An auto-continue happens after a given post-wait time, which you can edit by clicking in the post-wait column of the cue list. If a selected cue has an auto-continue, then a press of the GO button will start both the first cue and the post-wait. When the post-wait is complete, the second cue will start automatically. For example, say cue 12 is standing by, has a post-wait of 3 seconds, and cue 13 is next in the cue list. Pressing GO will play cue 12 immediately, wait 3 seconds, and then play cue 13, and it doesn't matter whether cue 12 has finished yet. By default, cues have a post-wait time of 0 seconds, which means both cues will start simultaneously.

To create an auto-continue, select *Auto-continue* from the drop-down menu in the bottom-left corner of the Basics tab in the inspector. An arrow with a dotted line will appear in the far-right column. To delete the auto-follow, select *Do not continue* from the drop-down menu.

Just as with auto-follow, you can click the dotted arrow to cycle through auto-follow, auto-continue, and do not continue, or use the assigned keyboard shortcut.

Start-all Groups

Cues in a start-all Group will start simultaneously when the Group cue is triggered.

To create a start-all Group, either create a Group cue through any of the available methods and drag-and-drop the cues that you want to include into the Group. Alternately, you can select the cues you want to include, and then create the Group cue.

The default keyboard shortcut to create a Group is ⌘**0**.

Once the Group cue is created, select it, then go to the Mode tab in the inspector and select *Start all children simultaneously*.

For more information about Groups, including about other types of Groups, please refer to the Group Cues section of this documentation.

Pre-wait

If a cue has a pre-wait time set, triggering that cue will start the pre-wait counter, and the cue will start when the pre-wait time has elapsed. You can assign a pre-wait to a cue by double-clicking in the *pre-wait* column of the cue list and entering a time. You can also use the keyboard shortcut **E**, or enter the pre-wait time in the Basics tab.

A pre-wait is a valuable tool to use in conjunction with start-all Groups as a way to create a timeline of sorts. For example, let's say you want a series of cues 20, 21, 22, and 23 to each start two seconds apart after pressing GO. You can put the cues in a start-all Group, and assign a pre-wait of two seconds to cue 21, four seconds to cue 22, and 6 seconds to cue 23. When the Group cue is triggered, all four cues will start simultaneously, but the pre-wait times of cues 21, 22, and 23 will elapse before they actually begin their action.

Disarmed Cues

Disarmed cues in a sequence do not interrupt or change the flow of events in a sequence. Their pre-waits and post-waits and follows are still respected, but the cue itself does not execute. Disarmed audio cues play no audio, disarmed video cues play no video, disarmed MIDI cues send no messages, and so on.

Licenses

License Types

QLab licenses come in two basic categories: permanent licenses and rental licenses. Permanent licenses last forever, whereas rental licenses have an expiration date set at the time of purchase.

Within both these categories, there are five types of licenses:

- Basic Audio
- Basic Video
- Pro Audio
- Pro Video
- Pro Bundle (Pro Audio and Pro Video licenses rolled into one and sold at a discount)

Please note:

- With the release of QLab 4, QLab 3 licenses are no longer for sale. You can buy a QLab 4 Audio or Video license, and use it with QLab 3.2 or newer. You can no longer purchase QLab 3 licenses.

Multiple licenses may be installed simultaneously within a single copy of QLab. You could, for example, own a permanent Basic Audio license and then rent a Pro Video license for just one week. Installing that Pro Video license will not affect your Basic Audio license at all.

Installing New Licenses

Installing licenses is most easily done if the Mac that's running QLab can be connected to the internet. (If your QLab Mac is offline, skip to the section below called *Offline Activation*.)

First, you need to make sure your copy of QLab is updated to version 3.2 or newer, which you can do by opening QLab and choosing *Check for Updates*... from the **QLab menu.**

Once you're updated, choose *Manage Your Licenses*... from the **QLab menu.** You will be presented with the Licenses and Activations window. Click *Log in to install a license*... and enter your email and password, or enter only your email and click *Get Link*.

Using your password

If you enter your password, QLab will present you with a list of licenses associated with your account, along with buttons to install the license.

Once you choose a license, you will be presented with another choice; which *seat* to install. Each QLab license can be installed on up to three Macs at a time, with three specific uses:

1. **Main.** The Main computer is used to run cues during your show. A show that uses one Mac for playing back Audio and another Mac for playing back Video has two Main computers.
2. **Backup.** The Backup computer is there in case something goes wrong with the Main computer. You might have your Backup computer hooked up to your playback system, with a system in place to seamlessly switch over from the Main (this is often called a "hot spare" or "hot backup"), or you might have your Backup computer disconnected, but ready to plug in at a moment's notice (called a "cold spare" or "cold backup.") In either case, the Backup doesn't do anything that's audible or visible to the audience during the show unless it takes over for the Main computer.
3. **Edit.** The Edit computer is used to work on your QLab workspaces in rehearsal, at your studio, or at home. If you have a visiting artist working on your show, you might loan your Edit seat to her so that she can work on her cues using her own laptop, rather than having to sit in the technical booth to work on cues. The Edit computer never does anything that's audible or visible to the audience.

Once you choose a seat, the Licenses and Activations window will show you the license and seat that you installed.

Using *Get Link*

If you click *Get Link*, we will send you an email containing a temporary license link which you can click to automatically log QLab in to your account. This makes it easy to install your license without having to remember your password, as long as you have access to your email on the same Mac that you're using for QLab.

Installing Preexisting QLab 3 Licenses

When you first purchased the license, received a license file via email along with your receipt. To install that license, first download QLab onto your computer (from http://figure53.com), then double-click on the license.

Occasionally, email servers (particularly Outlook servers) mangle the attachment. The file name of the license should look like this:

```
Pro_Bundle_v3_Standard_Q1029384756ZXCVASDF.QLab3License
```

If it does not, first try changing whatever suffix follows the dot in the filename to ".QLab3License" and then double-click it. Sometimes licenses come through with the suffix ".QLab3License.XML" and all you need to do is delete that ".XML" to make it work. If it still does not work, contact the support team and we'll help you out.

You can also install a license from inside QLab, by choosing *Other License Actions > Install a Legacy License File* from the **QLab** menu, and then selecting the license file and clicking *Open*.

Offline Activation

Adding and removing QLab licenses is typically done over the internet, and that's the fastest and easiest way to do it. If your Mac cannot be connected to the internet, maybe because you're on a boat, or your IT department is grouchy, you can still install a QLab license by using offline activation. Here's how to do it:

1. Install QLab on your offline Mac. You'll need to download it on a computer that is online, and then you can transfer it to the offline Mac via a local network, a USB key, or any other kind of external drive.
2. Launch QLab and choose *Other License Actions* → *Save Offline Activation Request...* from the **QLab** menu. You will be prompted to save the request file.
3. Take that file and email it to us at support@figure53.com, and let us know which license and seat you wish to install. For example, you might say "I want to install the Main Audio seat of my Pro Bundle" or you might say "I want to install the Main Lighting and Backup Video seats of my Pro Bundle."
4. We'll take the request file, hand it to our team of license-processing elves, and email you back with the activation file.
5. When you receive that email, copy the file to the offline Mac (again using a local network or a USB key or some similar method.)
6. Finally, in QLab, choose *Other License Actions* → *Import Offline Activation...* from the **QLab** menu, and select the activation file. Presto, change-o, the license will be installed!

Rental licenses

Rental licenses operate according to the same policy as permanent licenses. This is a change from how rental licenses operated in the past, and this change reflects our evolving understanding of the ways that rental licenses are used, and how that reflects the costs of developing and supporting QLab.

If you install a rental license before its start date, it will sit there patiently doing nothing until its start date comes along, at which point it will spring to life. You can install multiple rental licenses, each with different start dates, and they will

each activate on their start date and deactivate on their expiration date, all without interfering with each other.

Modernizing licenses

QLab 4 introduced a new method of licensing which is account-based, rather than being based on a license file. The account-based system allows more exact tracking of which Mac has a license installed, and will bring more advances over time. If you own a QLab 3 license, you can "modernize" this license and bring it into the account system if you wish, but you do not have to.

If you want to modernize a QLab 3 license, you must first deauthorize it on all computers, and you must have a QLab account which you can create here. You can then write to us at support@figure53.com and we'll migrate your license into your account.

You do not need to modernize a license in order to use it for an upgrade discount.

Removing licenses

To remove a permanent license, connect your Mac to the Internet, choose *Manage Licenses…* from the **QLab** menu, and click *Deauthorize and Delete this License*. It's important to do this before doing things like erasing and re-initializing the computer or selling the computer, because simply removing QLab will not deauthorize your license, and thus the Mac will still count towards your total of three installations.

Please note that **we cannot remotely deauthorize licenses** which have not been modernized. If your Mac is damaged or stolen and deauthorization is impossible, contact the support team and we can note this fact in our license database.

If you're using a QLab 4 license or a QLab 3 license that has been modernized, you can also remotely remove a license from any Mac by logging into your QLab account on our website. To do this, click on the *See details* link next to the license you want to remove, and then click *Deactivate* for the seat you want to deactivate. The link will change to say *Cancel pending deactivation*. If you click again, it will change back to *Deactivate* and nothing will change. If you leave it alone, the seat will be released the next time QLab is launched on that Mac. Please note that the Mac must be connected to the internet when QLab launches for remote deactivation to work.

In between the time that you click *Deactivate* and the time that QLab is relaunched, the seat will remain unavailable for reinstallation.

Features by License

Feature	Free	Basic Audio	Pro Audio	Basic Video	Pro Video
Channels of audio output	2	8	48	2	2
Channels of audio input	2	8	24	2	2
Channels of audio per file	2	8	24	2	2
Audio waveform view	✓	✓	✓	✓	✓
Unlimited slices per Audio or Video cue	✓	✓	✓	✓	✓
Sample-accurate playback sync	✓	✓	✓	✓	✓
Audio fades	✓	✓	✓	✓	✓
Live fade previews	✓	✓	✓	✓	✓
Edit output channel names		✓	✓		
Audio device routing		✓	✓		
Audio effects per cue			✓		
Audio effects on cue outputs			✓		
Audio effects on device outputs			✓		
Fade audio effects			✓		
Fade playback rate of Audio cues			✓		
Devamp cues		✓	✓	✓	✓
OSC cues		✓	✓	✓	✓
MIDI cues		✓	✓	✓	✓
Script cues		✓	✓	✓	✓
Timecode cues			✓		✓

Feature	Free	Basic Audio	Pro Audio	Basic Video	Pro Video
Remote control via OSC, MIDI, iOS app	✓	✓	✓	✓	✓
One single-screen surface	✓	✓	✓	✓	✓
Unlimited single-screen surfaces				✓	✓
Unlimited multi-screen surfaces					✓
1000 layers of video	✓	✓	✓	✓	✓
Full-screen video cues	✓	✓	✓	✓	✓
Custom geometry video cues				✓	✓
Fade and animate video cues				✓	✓
Fade playback rate of video cues					✓
Surface masking					✓
Edge blending					✓
Warping and keystone correction					✓
Video effects					✓
Syphon input & output					✓
Blackmagic device input & output					✓
Camera cues					✓
Titles cues				✓	✓
Unlock QLab 2 Pro Audio			✓		
Unlock QLab 2 Pro Video					✓

Templates

Templates are a way to save entire workspaces as starting points for future work.

To create a template, make a new workspace and adjust all its settings to suit your preference. You can also create cues or cue lists which will serve as jumping-off points. For example, you have a set of Script cues that you routinely install into every workspace, you can include those too.

Save this workspace as a template by choosing *Save As Template* from the **File** menu. Pick any name you like. If you choose the same name as an existing template, you'll be asked if you want to replace the existing template.

To create a new workspace using your template, choose *New From Template* from the **File** menu, or use the keyboard shortcut ⇧⌘**N**. The Template Chooser will appear, and you can select the template you wish to use. Once you do, a new workspace will open up which is an exact copy of the template as you saved it.

To delete or rename templates, choose *Manage Templates* from the **File** menu.

QLab Remote

QLab Remote is an iOS app which can be used in tandem with QLab 3 running on a Mac on the same WiFi network. It can be used to view cues, cue lists, and active cues, and edit some aspects of cues.

Quick Reference

- **Long press on cue.** Move playback position to this cue.
- **Double tap on cue.** View details of this cue.
- **Tap on Flag icon.** Flag or unflag this cue.

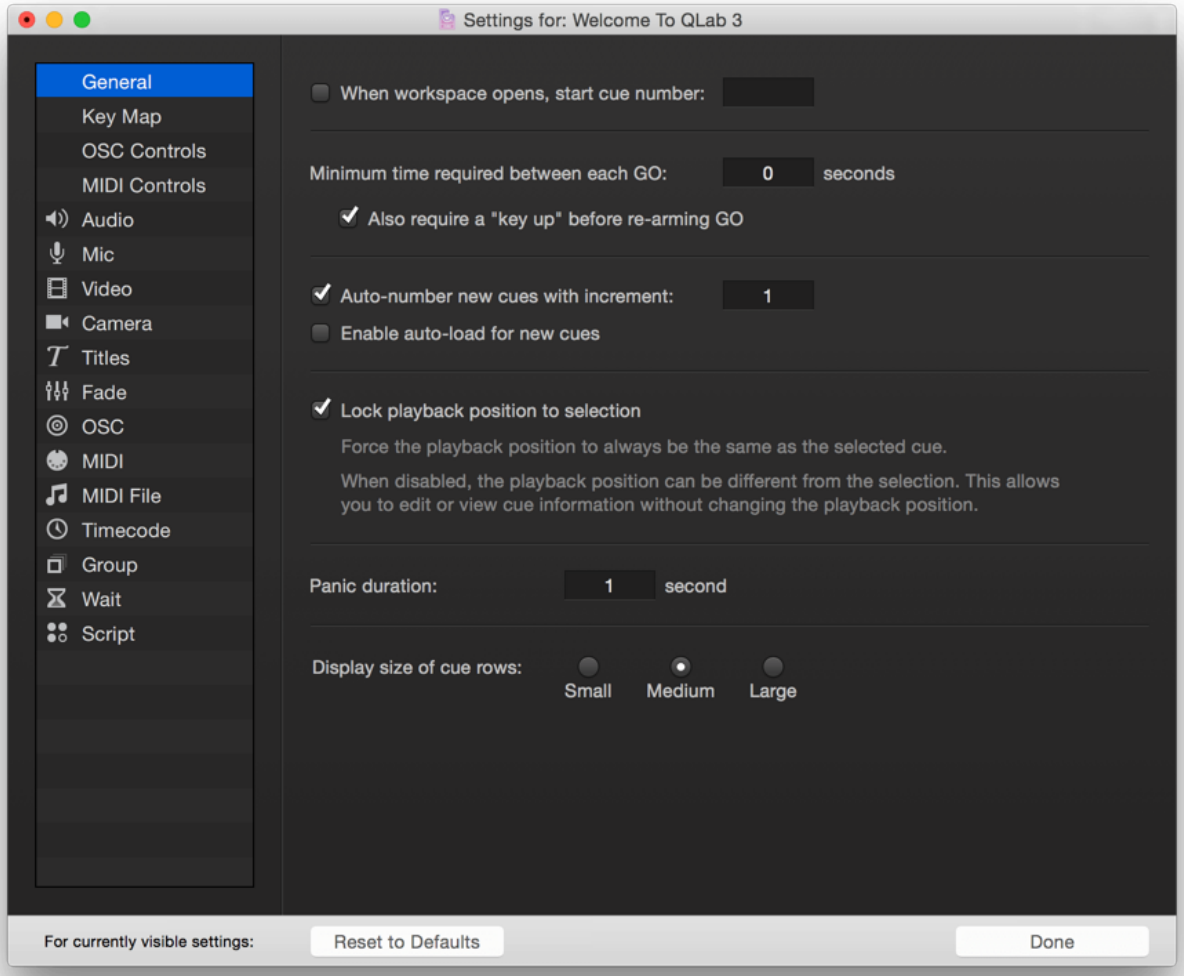
Troubleshooting

If QLab Remote is not showing any workspaces to connect to, ensure that:

1. Your iOS device and Mac are on the same WiFi network.
2. The workspace you are trying to connect to has OSC Controls enabled.
3. The WiFi network you're using allows access to TCP and UDP ports 53000 and 53001.

Settings

General



When workspace opens, start cue number. Check this box and enter a cue number in the text field, and QLab will automatically start that cue when this

workspace is opened. If you need to temporarily prevent this cue from starting, hold down the **control** and **option** keys (⌃⌥) while opening the workspace.

Minimum time required between each GO. This can be set to any duration (including 0) in order to help prevent accidental starting of cues due to a double-press of the GO button. This protection will apply to mouse clicks on the GO button, pressing **space** or any other key you've assigned to GO, and incoming MIDI, MSC, and OSC controls which directly trigger the "GO" action.

Also require a "key up" before re-arming GO. This protection applies only to the keyboard shortcut assigned to GO; it forces QLab to wait for the key to be released before accepting another command to GO.

Auto-number new cues with increment. Check this box and enter a number into the text field to automatically assign cue numbers that increase by the desired increment to newly created cues. Cues will be numbered with the lowest available number according to this increment.

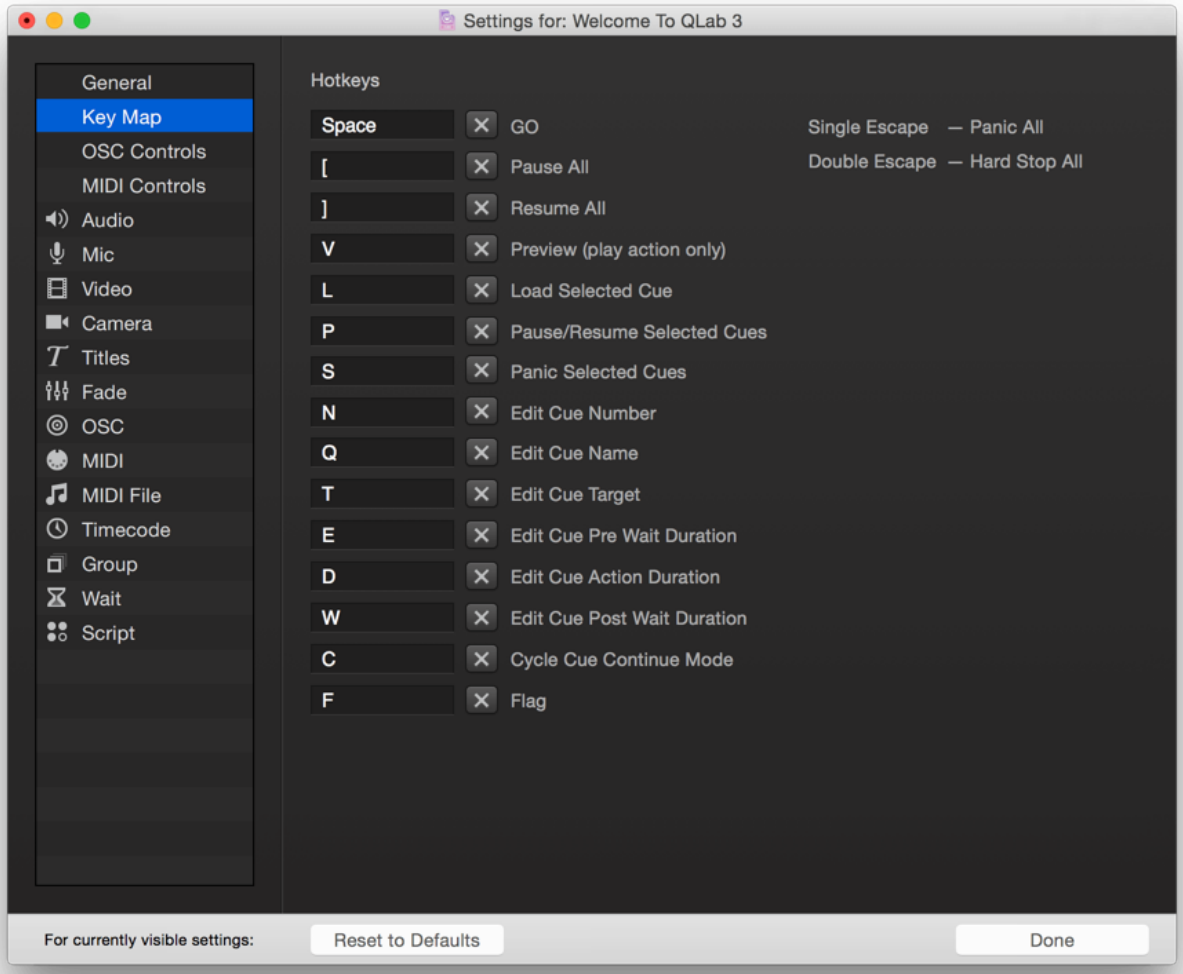
Enable auto-load for new cues. Check this box and newly created cues will have their "auto-load" option enabled by default.

Lock playback position to selection. Check this box to make the playback position always match the selected cue. When disabled, the playback position can be different from the selection. This allows you to edit or view cue information without changing the playback position.

Panic duration. This is the time over which all running cues will fade out and stop when the escape key is pressed or the cue list receives a command to panic.

Display size of cue rows. Choose a size to display your cues: small, medium, or large. Small allows more information to fit on screen, and large is convenient for easier viewing from a distance.

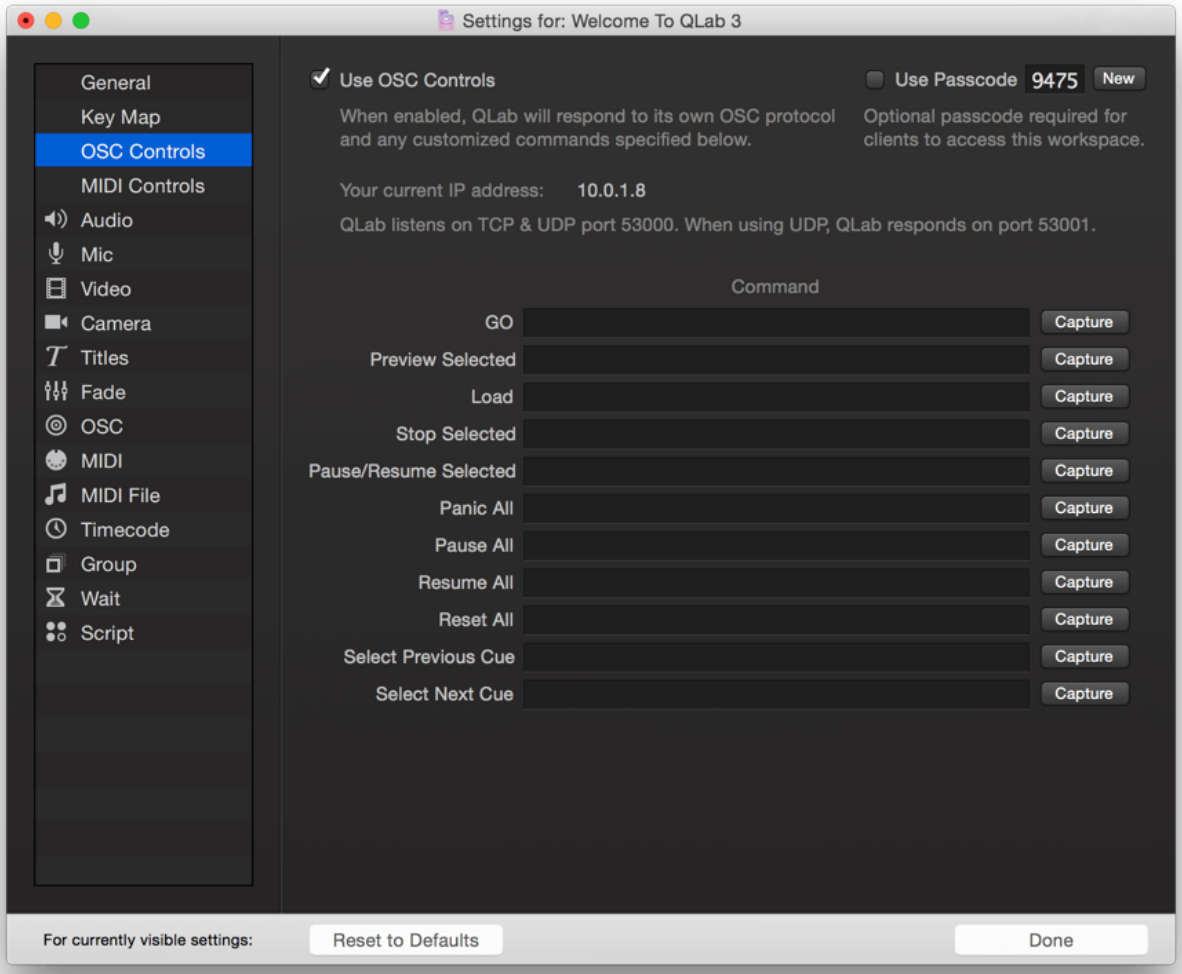
Key Map



The Key Map allows you customize keyboard shortcuts, or hotkeys, for the listed commands. Simply enter the desired hotkey in the text field next to the appropriate command. To disable a hotkey, press the X button next to the command.

Note that the **escape** key is always assigned to the panic command.

OSC Controls



Use OSC Controls. When enabled, QLab will respond to its own OSC protocol as well as any customized commands for the transport controls listed below.

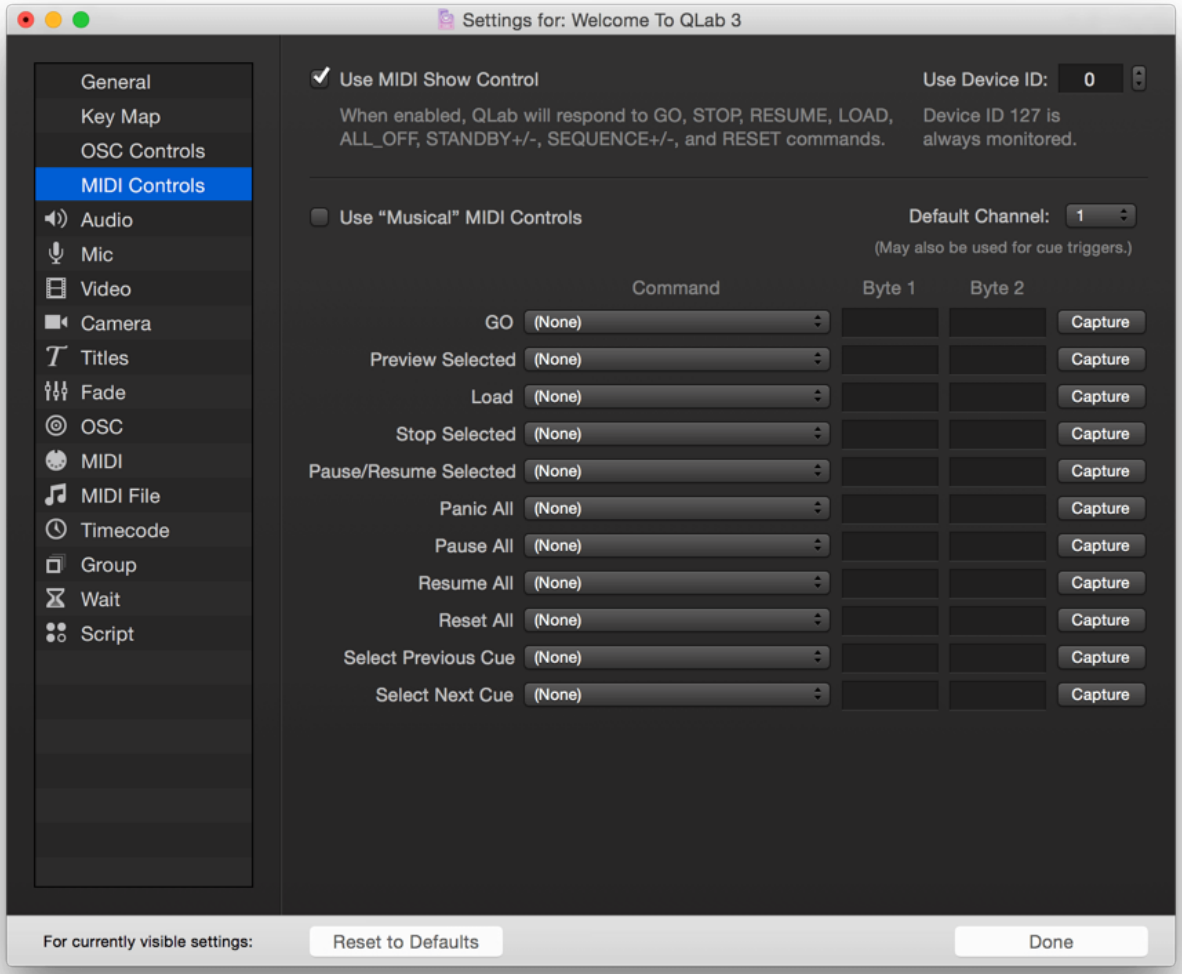
Use Passcode. Select this optional feature to require a four-digit passcode which clients must enter in order to send OSC commands to QLab.

Command. For the listed transport controls, you can manually enter any valid OSC command, or click “Capture” to have QLab listen for the next incoming OSC

command and assign that command to the selected control. Note that QLab's own OSC commands for these controls are not overridden by entries here; they remain active in addition to them.

QLab listens for OSC on all active network interfaces on port 53000. Additionally, QLab listens for plain text formatted as OSC commands on port 53535. When incoming commands are sent to port 53000 via UDP, QLab will send replies on port 53001.

MIDI Controls



Use MIDI Show Control. When enabled, QLab will respond to GO, STOP, RESUME, LOAD, ALL_OFF, STANDBY+/-, and RESET commands sent via MIDI Show Control. QLab listens for messages categorized as Lighting (General), Sound (General), and Video (General).

Use Device ID. Device ID 127 is always monitored, along with whatever ID is entered here. All devices on an MSC network should have unique device IDs.

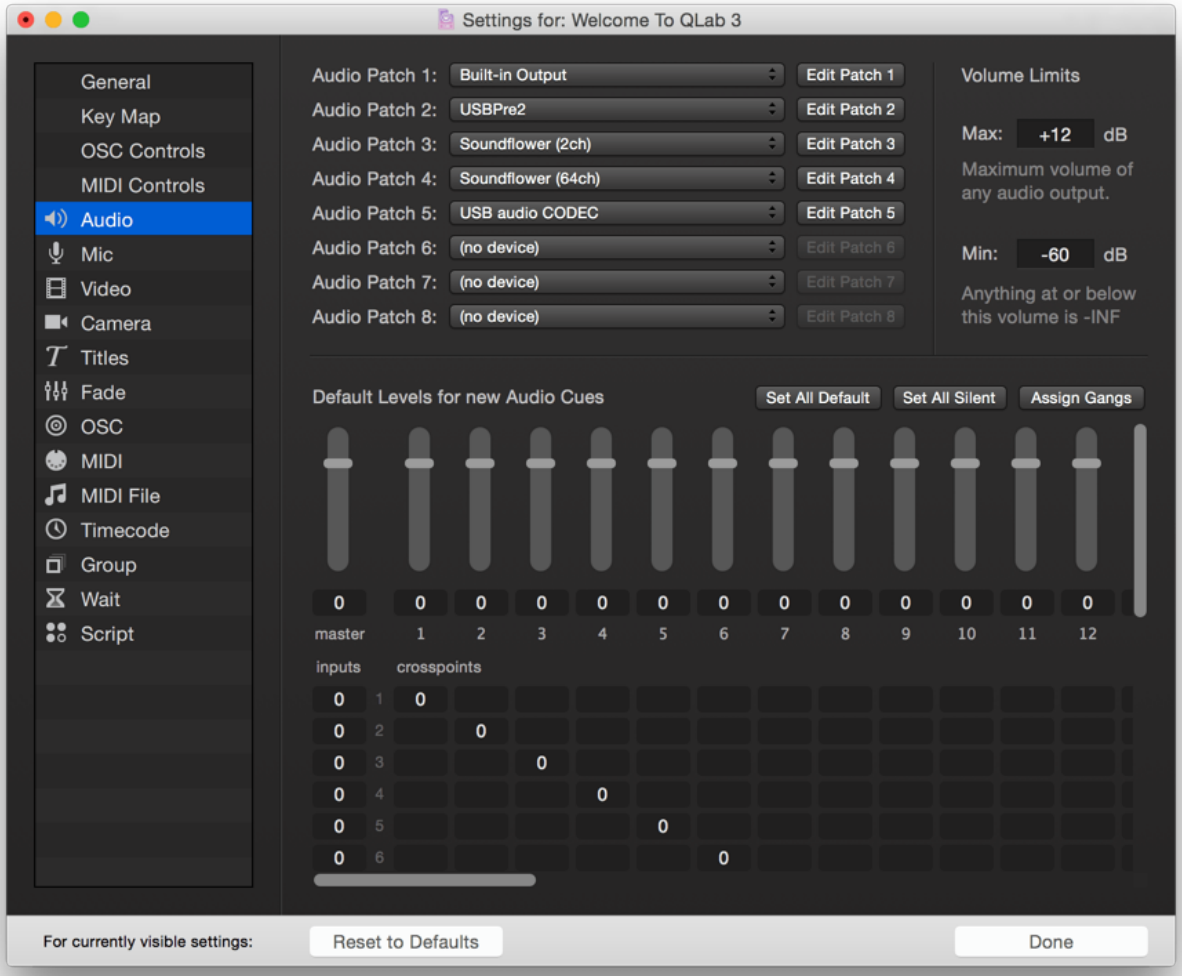
Use "Musical" MIDI Controls. When enabled, QLab listens for incoming MIDI messages which can be assigned to the transport controls listed below, as well as to individual cues as MIDI triggers.

Default Channel. You can restrict incoming MIDI to a specific channel for transport commands, or set to “Any.” Individual cue triggers will default to listening on that same channel, but can be individually set to a different channel.

Command. For the listed transport controls, you can manually enter a MIDI command, or click “Capture” to have QLab listen for the next incoming MIDI command and assign that command to the selected control. Note that only Note On, Note Off, Control Change and Program Change messages can be used for transport controls.

The field marked “Byte 2” can contain greater-than (>) or less-than (<) signs, and can also contain the word “any.”

Audio



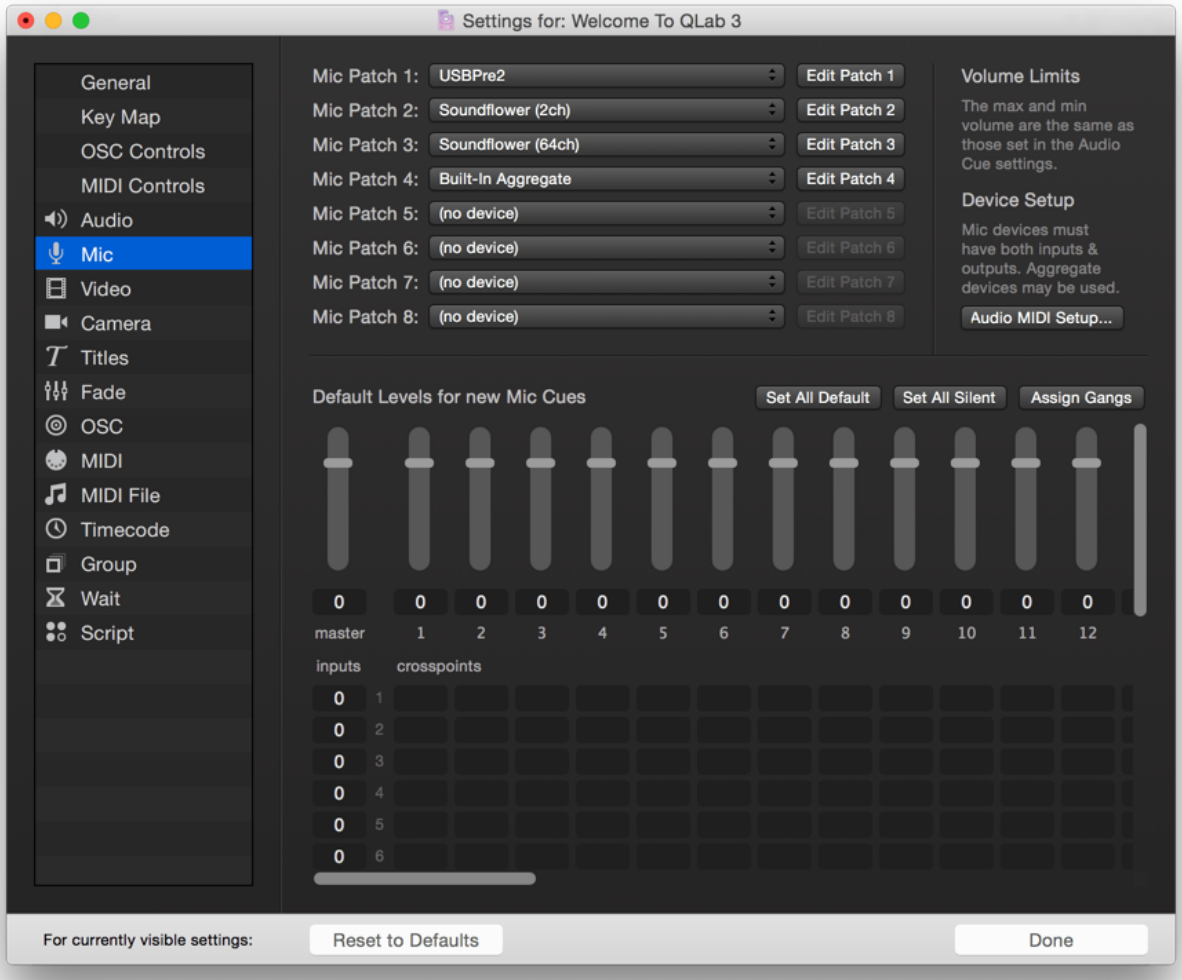
Audio Patches. QLab workspaces can connect to eight individual audio devices, which are assigned to one of eight audio patches. Use the drop-down menus to assign a new device to a patch. Click the *Edit Patch* buttons to the right of each drop-down menu to edit the details of that audio patch.

You can read more about editing audio patches here.

Volume Limits. Set an upper limit to prevent a sound from being played too loudly. Specify a lower limit, and QLab will treat anything at or below that level as `-INF` (silent).

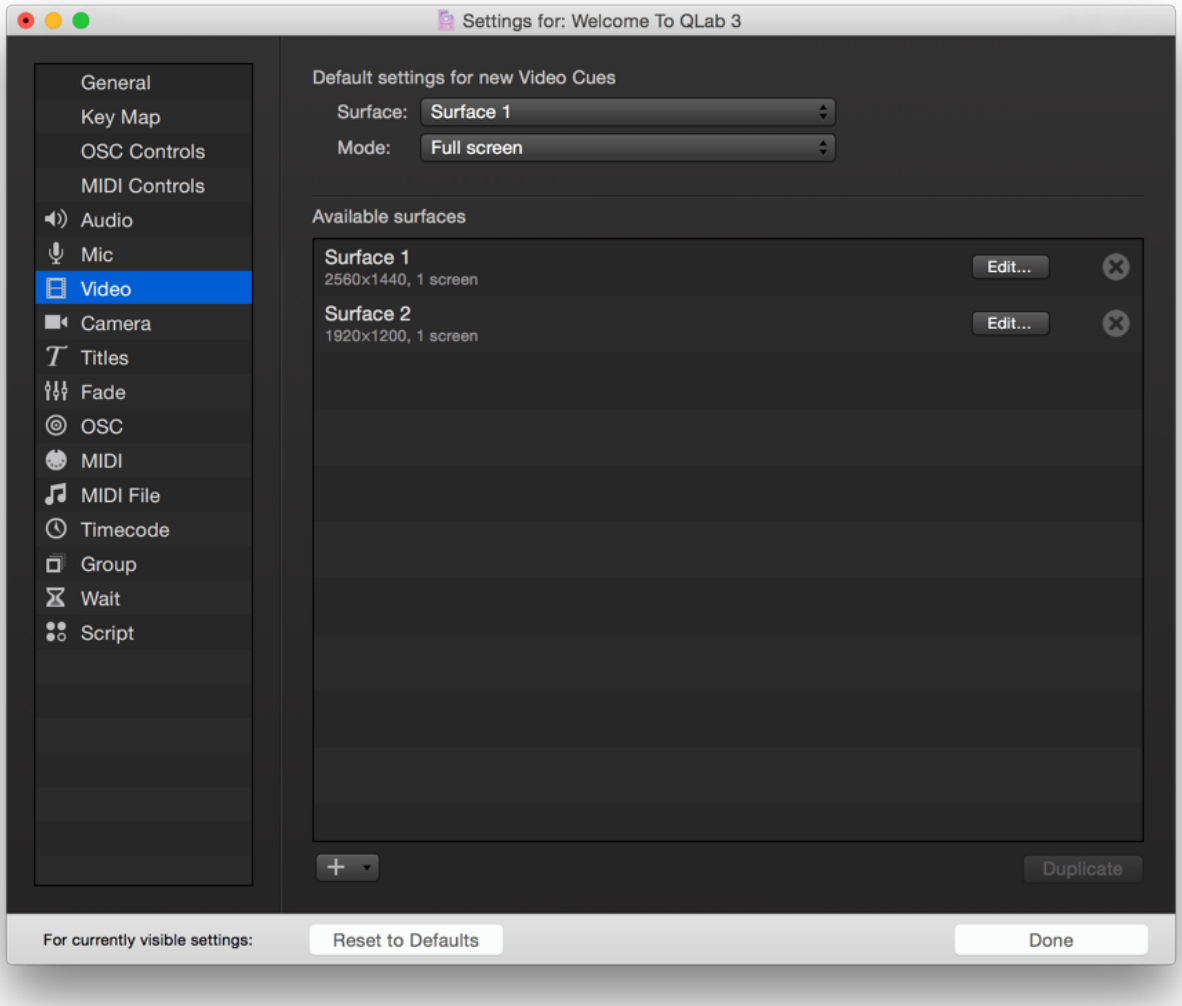
Default Levels for new Audio Cues. The levels in the matrix mixer are used as defaults for all new Audio cues, and for the audio levels in new Video cues. Adjust as desired by moving the sliders up and down or by entering a value in the text field. There are buttons for resetting the matrix to QLab's built-in defaults, for setting all controls to `-INF` (silent), and for assigning default gangs.

Mic



This section is essentially identical to Audio settings. Patches and defaults need to be set separately for Mic cues, and can be either the same as or different than those for Audio cues.

Video



Default settings for new Video Cues. You can set the default Surface and Mode (full screen or custom geometry) for newly created Video cues.

Available surfaces. The surfaces defined for this workspace are listed here to be edited or removed. You can learn more about editing surfaces here.

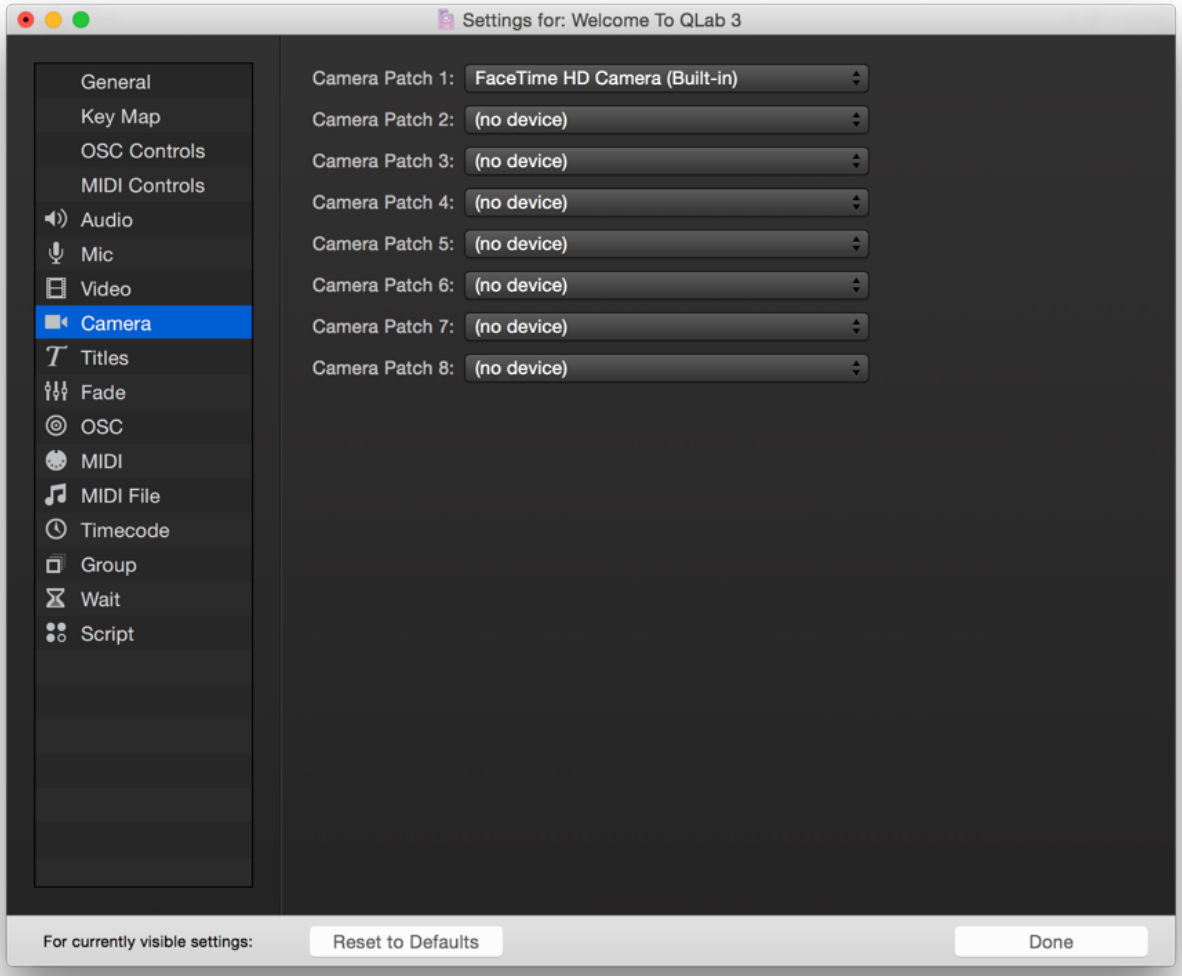
By default, in a new workspace, QLab creates one surface for each display connected to your Mac.

The + button below the list of surfaces gives you three ways to create a new surface.

- *New Empty Surface* creates a new surface with no screens assigned to it.
- *New With Display* creates a new surface with the selected screen or partial screen pre-assigned, and with dimensions equal to that screen's.
- *New Multi-Screen Surface* creates a new surface via a helper tool which lets you enter the resolution of each projector, the physical layout of the projectors, and the percentage overlap between each projector.

Duplicate makes a copy of the selected surface.

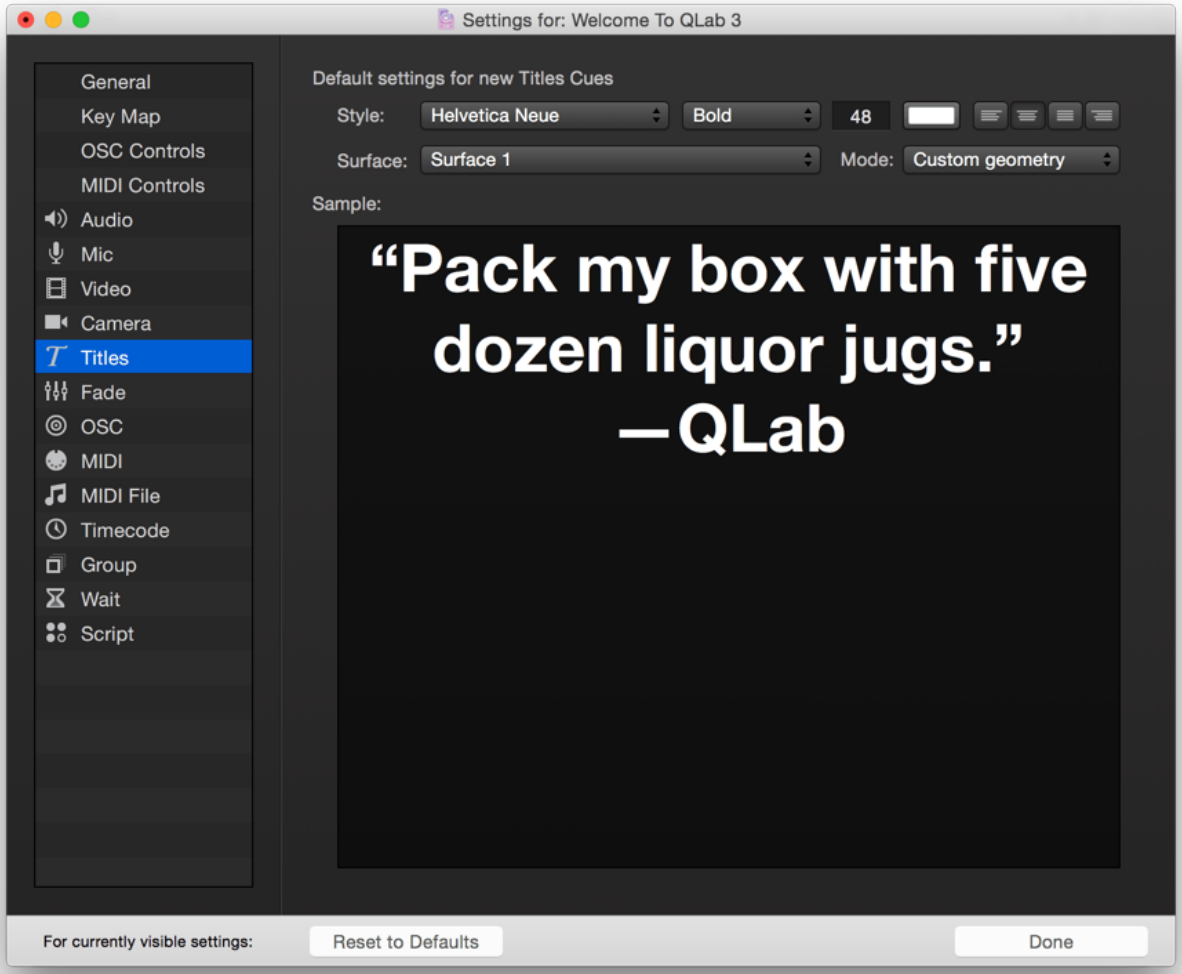
Camera



Camera Patches. QLab workspaces can connect to eight individual video input devices, which are assigned to one of eight Camera Patches. Use the drop-down menus to assign a device to a patch.

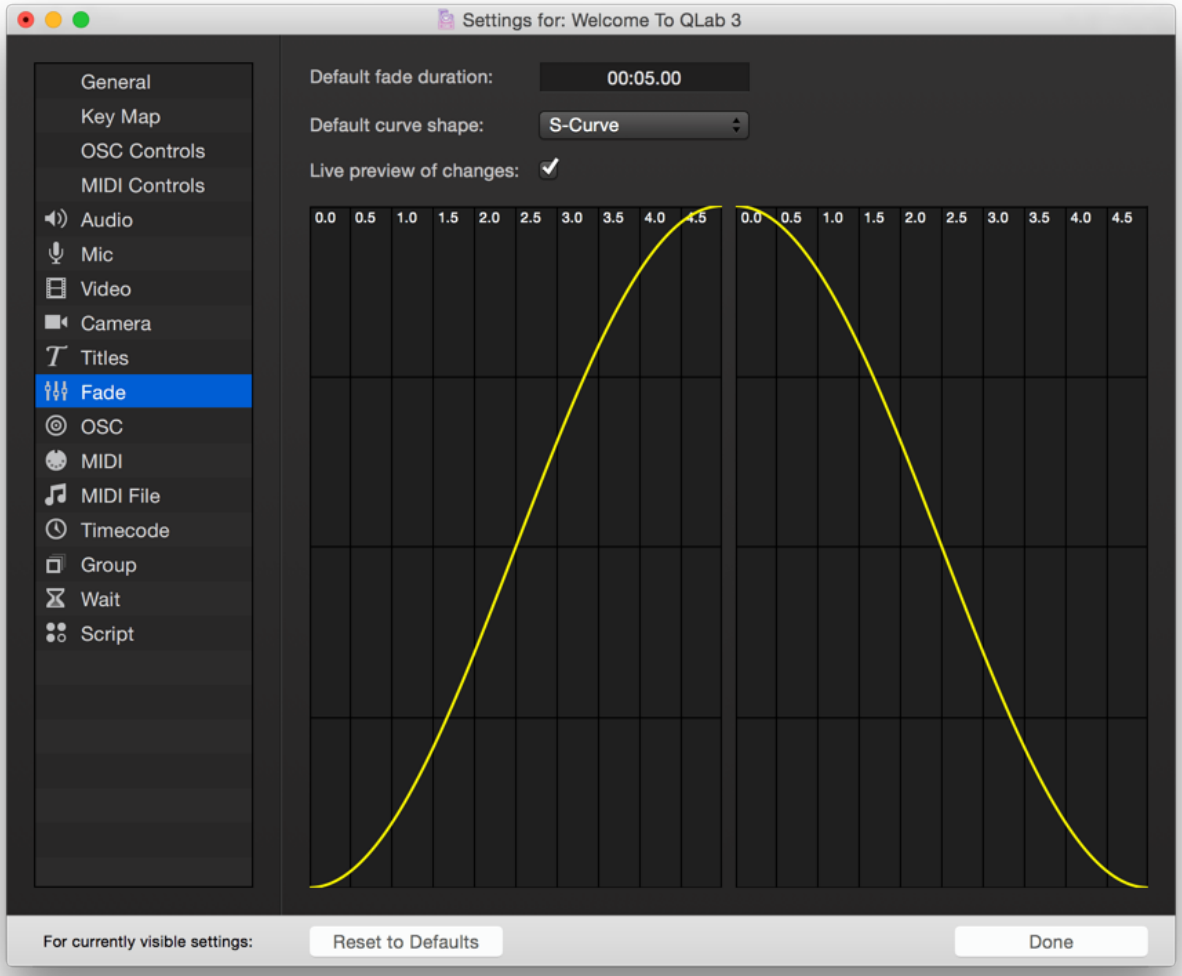
Camera cues in QLab can use the iSight or FaceTime camera built into many Macs, most USB- or FireWire-connected webcams, and Blackmagic video capture devices belonging to the UltraStudio, Intensity, DeckLink, and MultiBridge product lines.

Titles



You can set the default font, style, size, color, justification, surface, and mode for new Titles Cues.

Fade

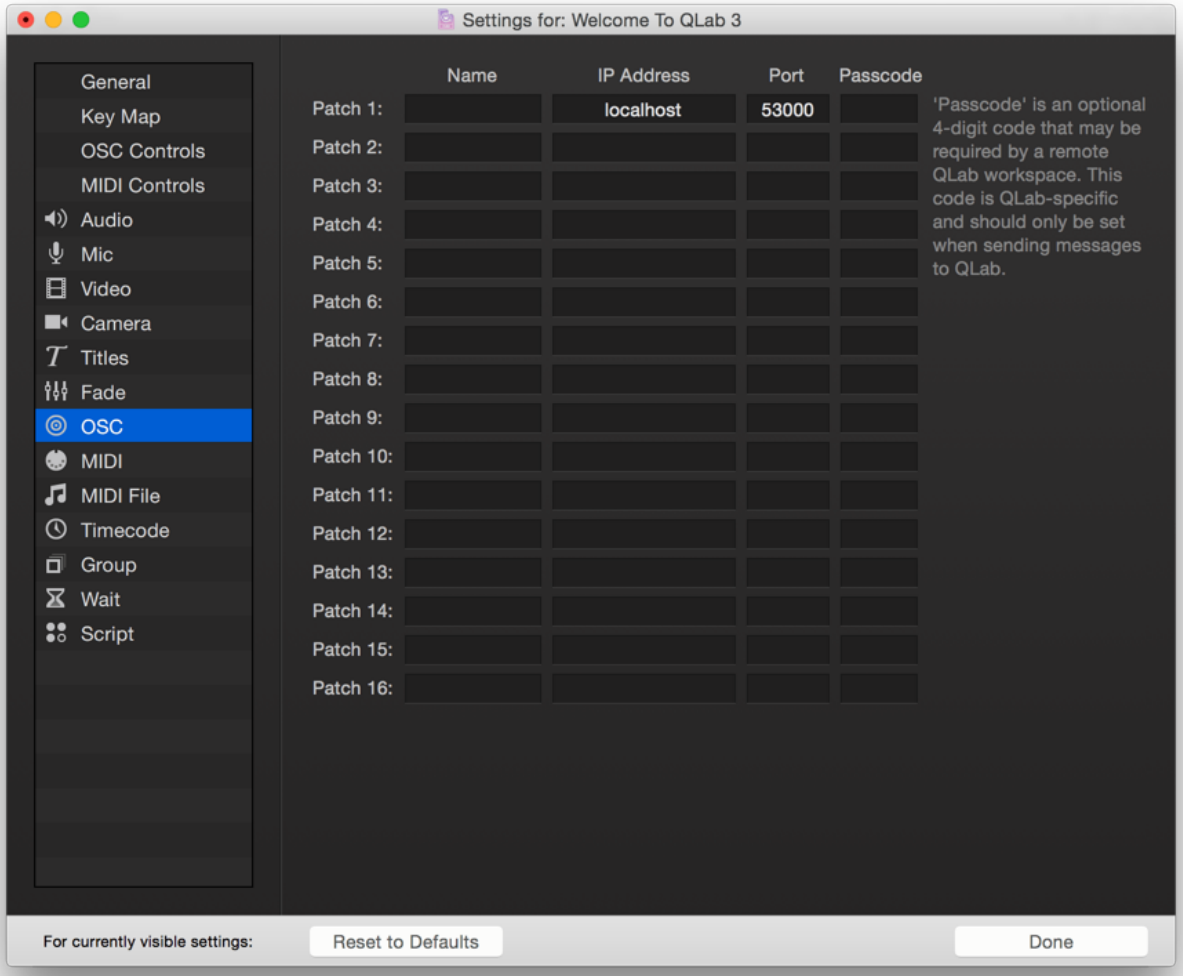


Default fade duration. The fade time for newly created Fade cues.

Default curve shape. Choose between S-Curve and Custom Shape. If you choose Custom Shape, you can define the shape separately for up-fades and down-fades.

Live preview of changes. When enabled, any manual change made to a Fade cue will be reflected in real time, assuming the target of the Fade is playing.

OSC



A workspace can send OSC messages to sixteen different destinations, set in the sixteen patches in the OSC preferences screen.

Name. This is for setting an easy-to-read label, and has no bearing on the functionality of the patch.

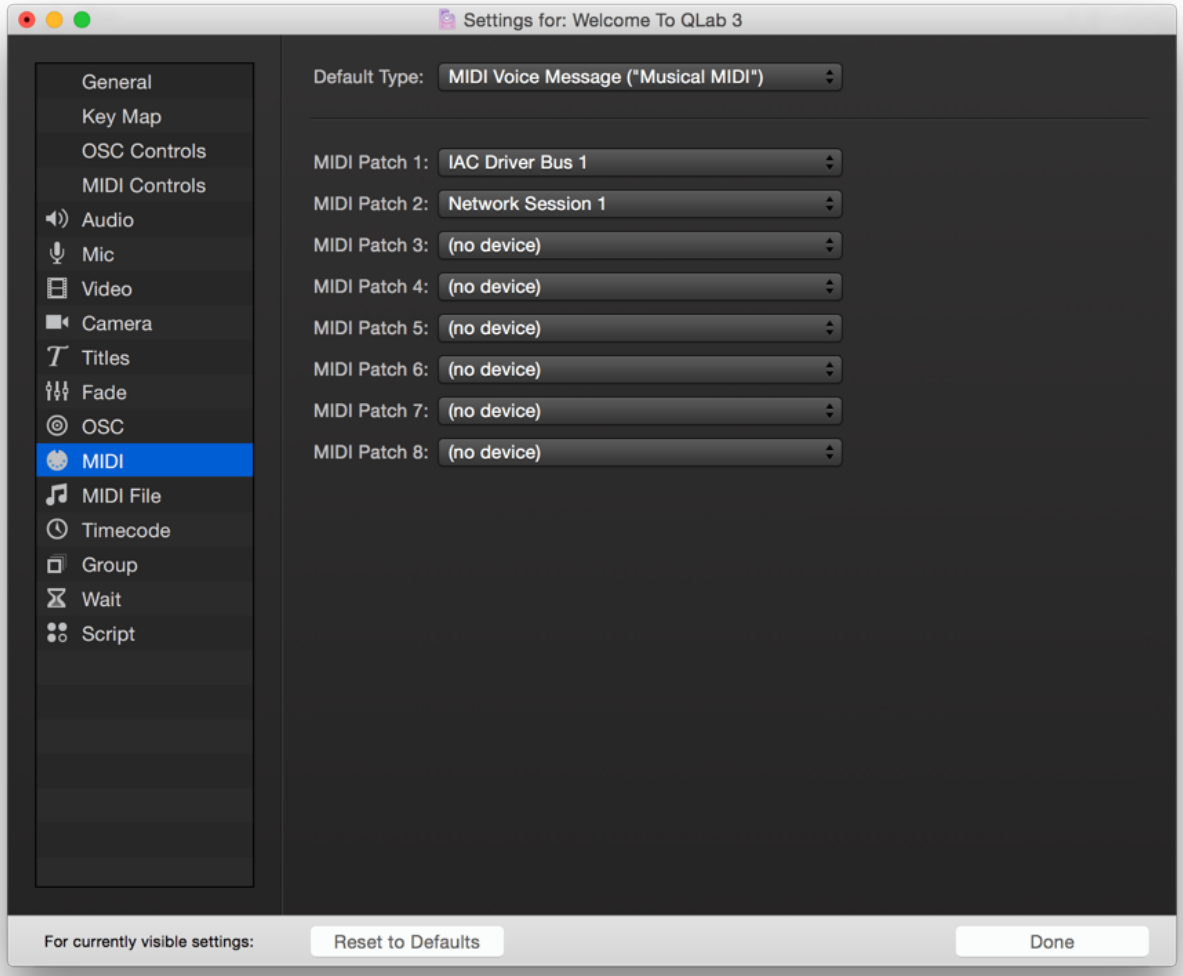
IP Address. This is the IP address of the destination. Typing *localhost* or *127.0.0.1* will set the destination to this computer, to allow QLab to send OSC messages to itself or to other software on the same machine.

Important: starting in QLab 3.1.15, in the interest of speed, messages sent to *localhost* do not go through the network stack of the Mac OS, and as such can only be used for sending OSC messages from QLab to itself. Messages sent to *127.0.0.1* do traverse the network stack, and as such can be used to send OSC messages to other programs running on the same Mac.

Port. This is the UDP port on which the destination device or software is listening. QLab listens for OSC on port 53000, and for plain text formatted as OSC on port 53535.

Passcode. If the destination is a copy of QLab, and that copy has a passcode set (see OSC Controls, above) enter that passcode here. This code is QLab-specific and should only be set when necessary.

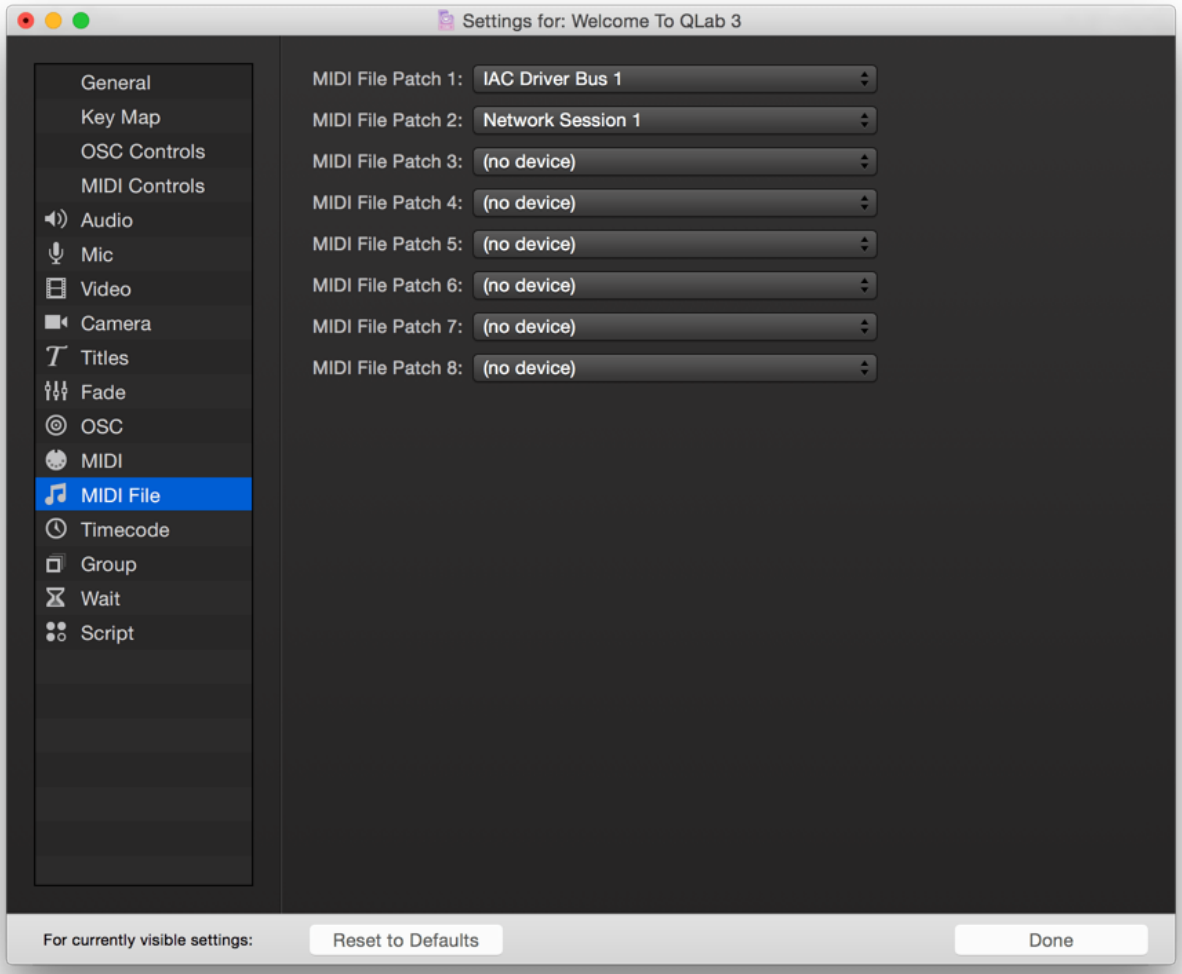
MIDI



Default Type. Select the default type of MIDI message for newly created MISI cues: MIDI Voice Message (“Musical MIDI”), MIDI Show Control (MSC), or MIDI SysEx Message.

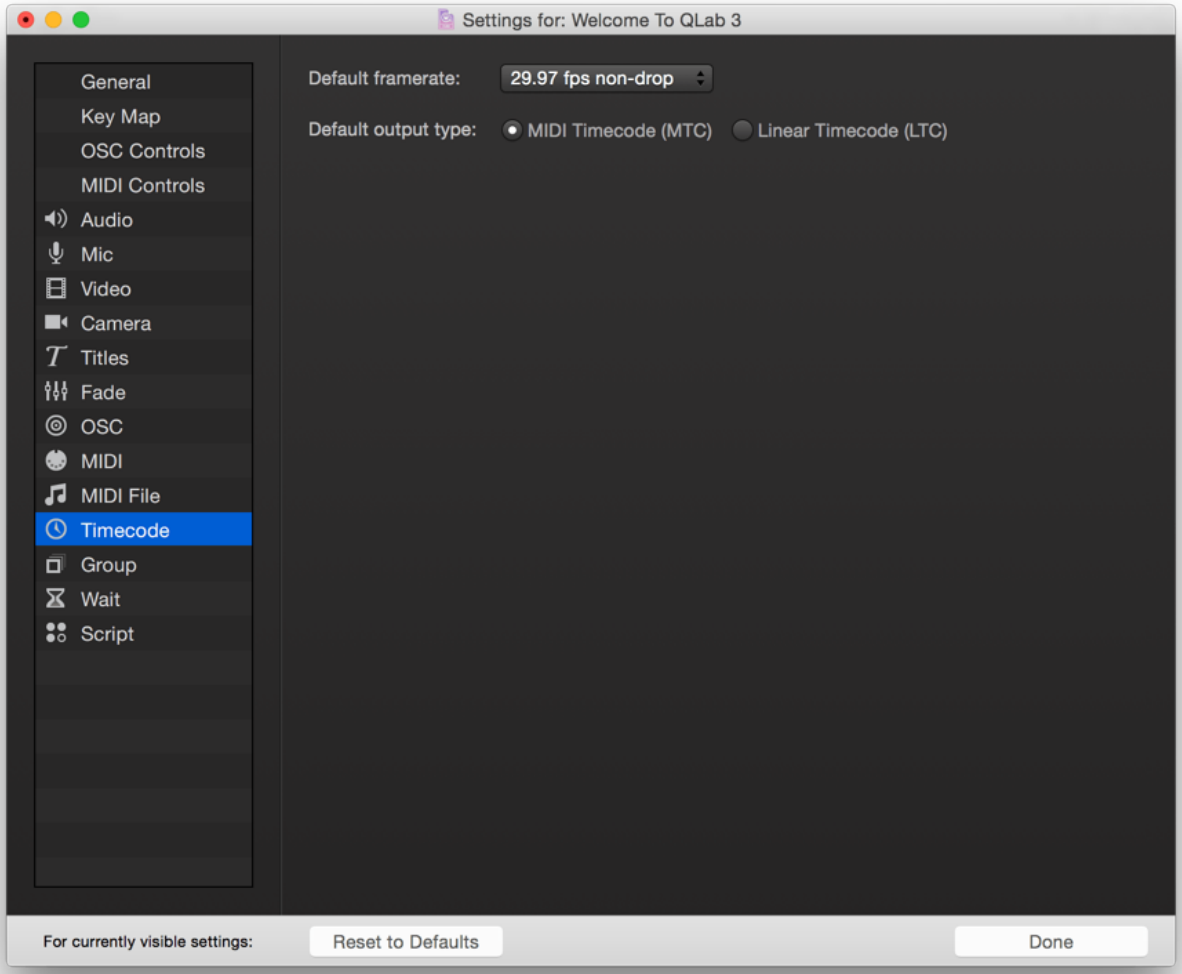
MIDI Patches. QLab workspaces can connect to eight individual MIDI devices, which are assigned to one of eight MIDI Patches. Use the drop-down menus to assign a MIDI device to a patch. Note that these patches are only relevant to outgoing MIDI from MIDI cues, and have no bearing on incoming MIDI for triggers or workspace controls.

MIDI File



This section is essentially identical to MIDI settings. Patches need to be set separately for MIDI File cues, and can be either the same as or different than those for MIDI cues.

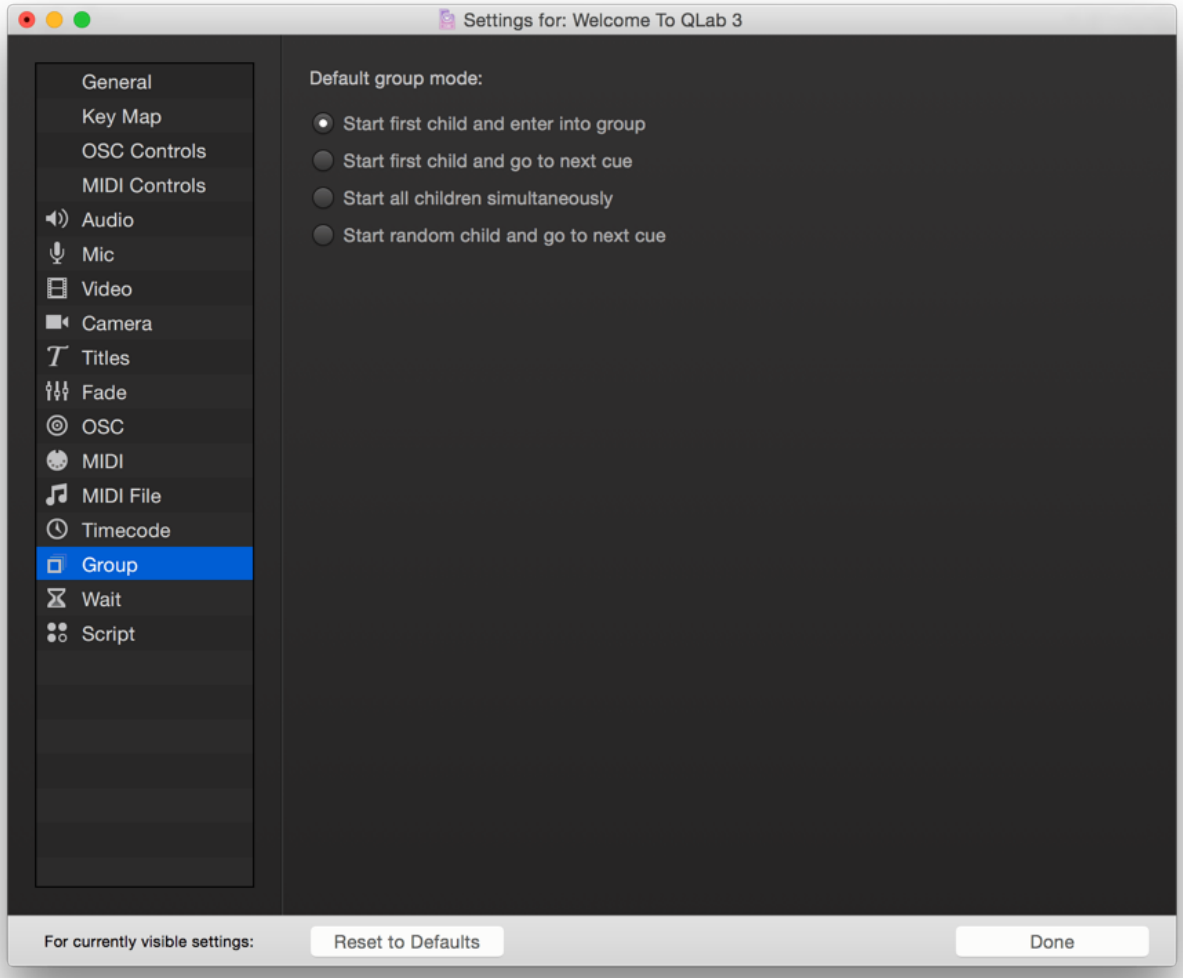
Timecode



Default framerate. Set the default framerate for newly created Timecode cues here.

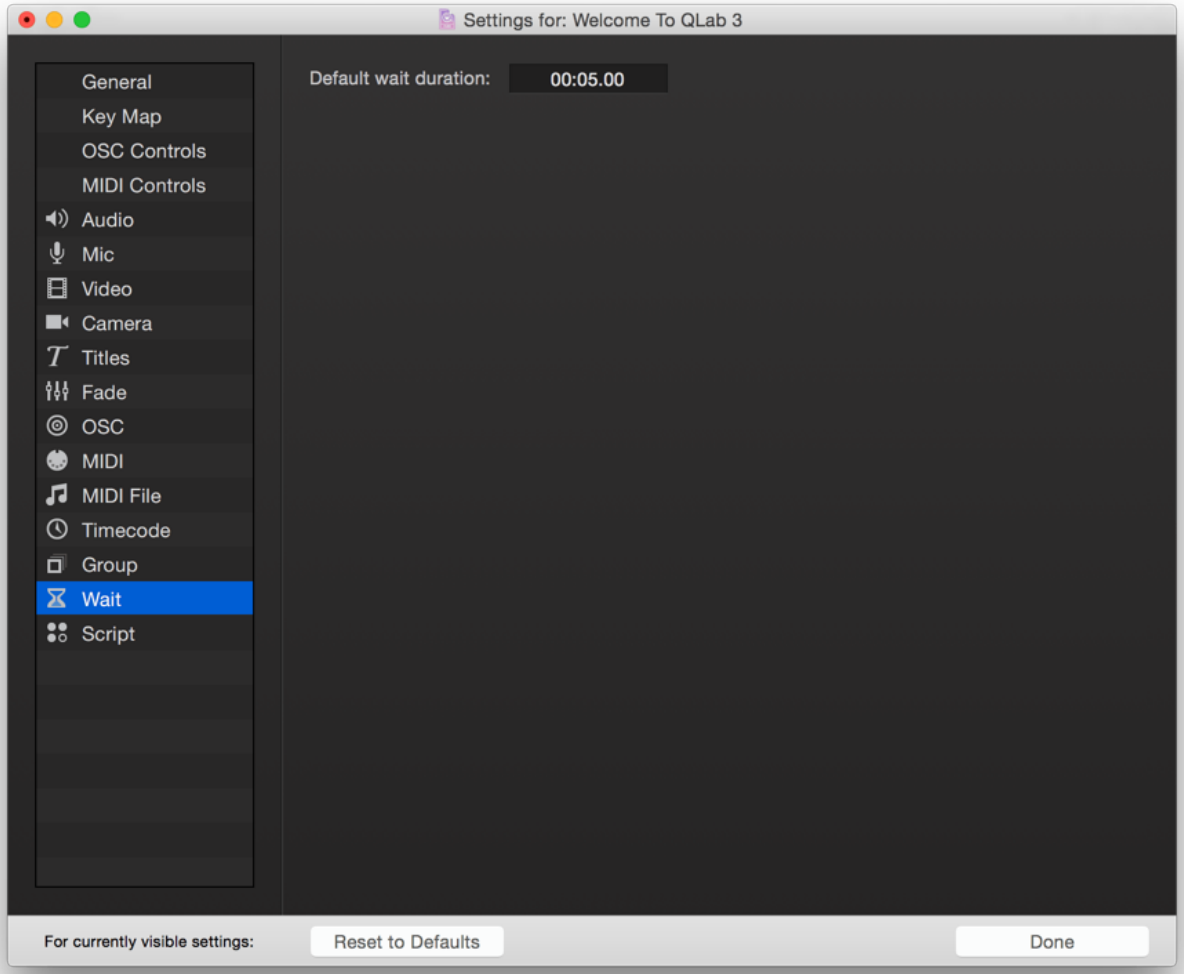
Default output type. Set the default type, either MIDI Timecode (MTC) or Linear Timecode (LTC, also sometimes referred to as SMPTE), here.

Group



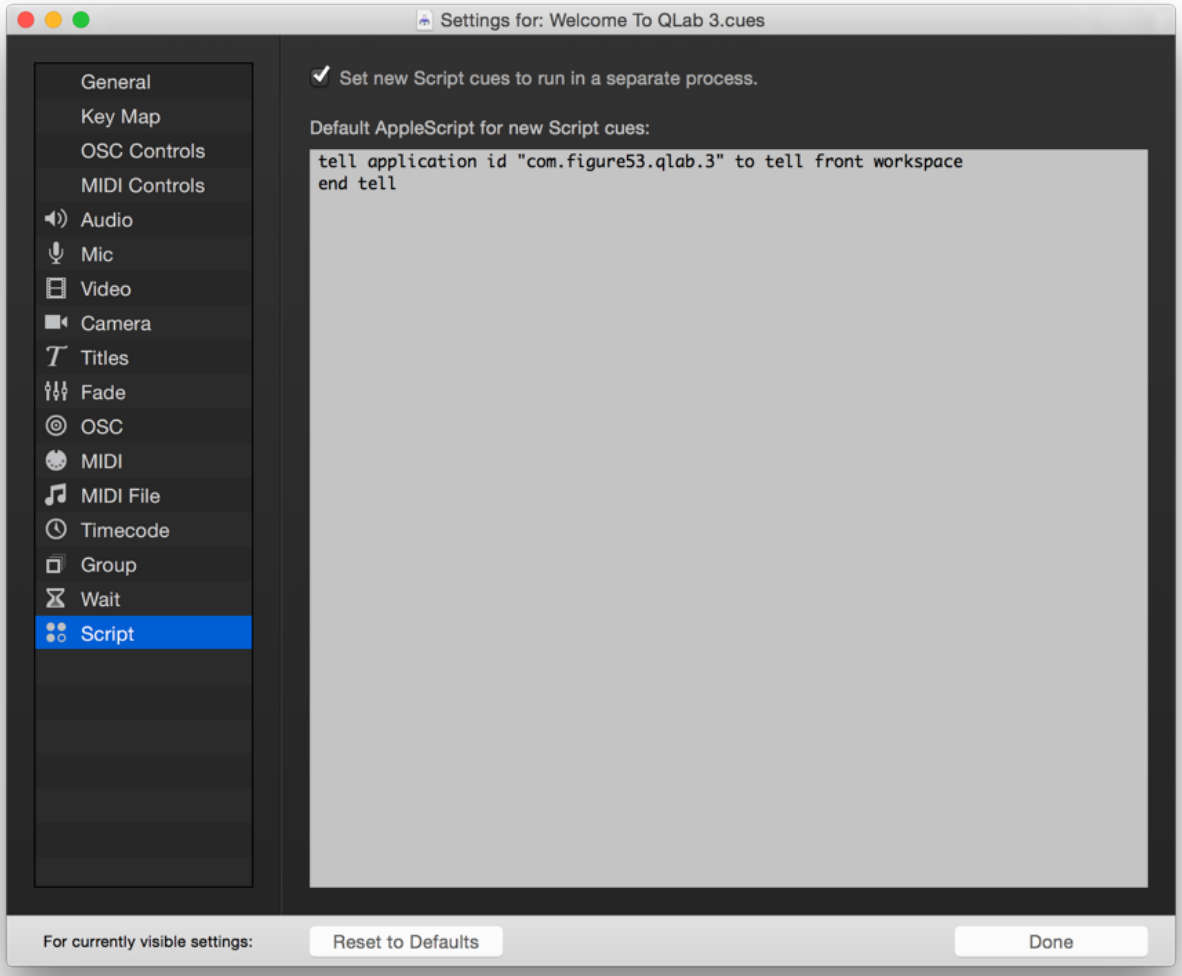
Default group mode. Set the default mode for newly created Group cues here.

Wait



Default wait duration. Set the default duration for newly created Wait cues here.

Script



Set new Script cues to run in a separate process. Running scripts in a separate process improves QLab's performance while the script is running, but prevents QLab from tightly synchronizing with that script. For complex scripts or scripts that don't have an element of synchronization to them, we recommend keeping this box checked.

Default AppleScript. Set the default contents of newly created Script cues here.

Chapter 2:

2.1 Audio

Audio

Chapter 3: Audio

Audio Cues

download video ↓

Audio cues require a target, which must be a file containing sound. QLab can play sounds in most formats, including:

- AIFF
- WAV
- CAF
- AAC
- MP4
- MP3 (this is QLab's least favorite format. Not only does it sound less than ideal, but it also has inherent timing problems that can interfere with critical timing in unpredictable ways.)

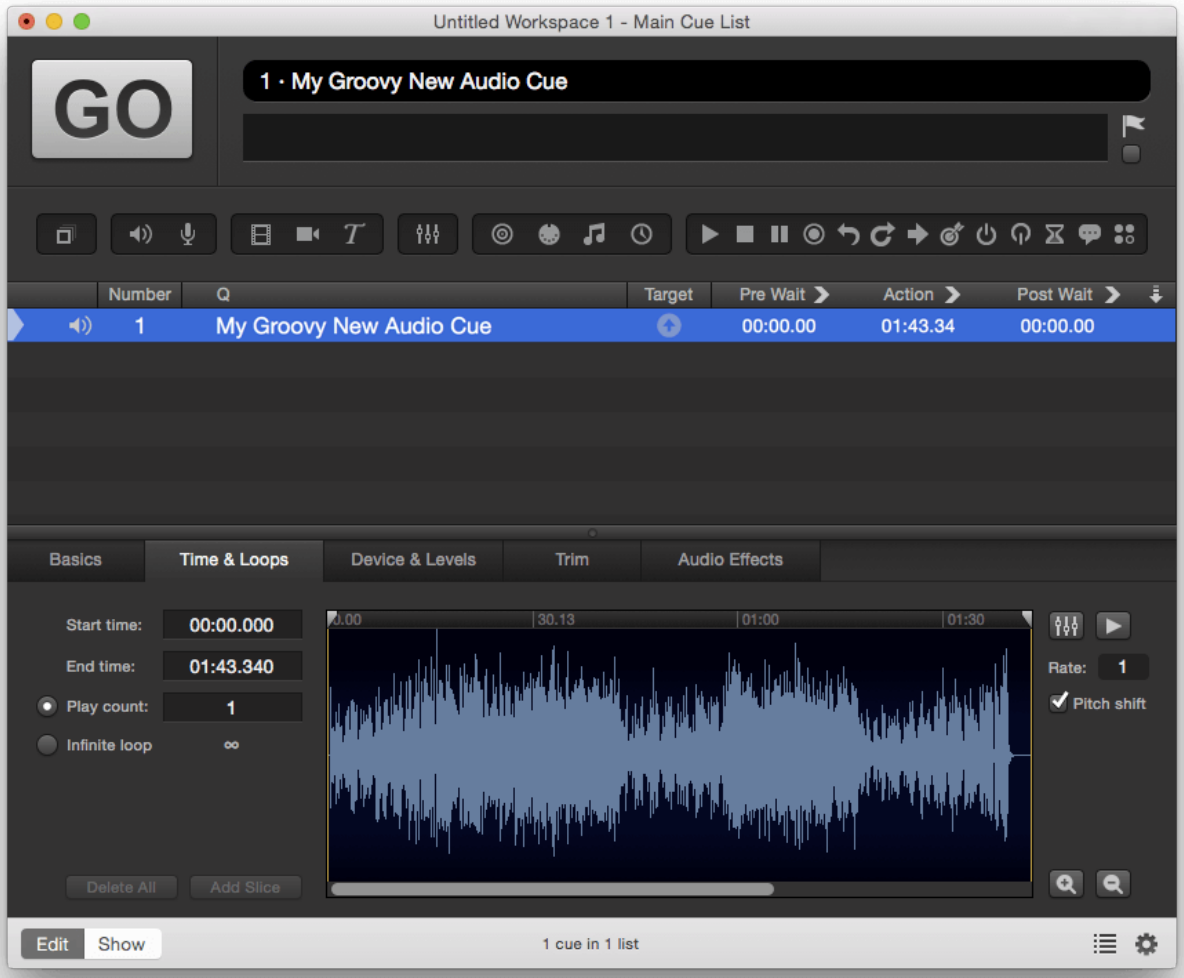
When a Audio cue is selected, five tabs will appear in the Inspector:

- Basics
- Time & Loops
- Device & Levels
- Trim
- Audio Effects

Basics

Please refer to the section on the inspector in the **Getting Started** section of this documentation.

Time & Loops



Start time and end time

Set the start and end times either by typing values into their respective text fields, or by dragging the time handles (downward-pointing grey triangles) within the waveform viewer.

Play count and infinite loops

Play count defaults to 1, but can be changed to any whole number (X) to repeat the cue X number of times. Alternately, select *Infinite loop* below the play count field to repeat the cue indefinitely.

Slices and the waveform view

The blue waveform on the right side of the *Time & Loops* tab is a visual representation of the audio cue. Scroll up and down on the waveform view to zoom in and out, or use zoom buttons on the bottom-right side of the tab.

To loop specific sections of an audio cue, instead of the whole thing, you will need to divide the track into slices. To create a slice, click in the waveform view and then click the *Add Slice* button to the left of the waveform. A marker similar to the start and end handles will then appear delineating the slice; these markers are green as opposed to grey. Click the handle at the top of the slice marker and slide it back and forth along the waveform to adjust its position, or click on the handle and then enter a value manually in the text field that appears on the left. Note that slices cannot be closer together than .1 seconds.

You can also click and drag within the waveform to highlight and select a section of the track, which will cause the *Add Slice* button to place slice markers on both sides of the selection.

If you bring in an AIFF or WAV file with preexisting markers, those markers will automatically appear as slices. Markers closer together than .1 seconds will be discarded by QLab, though the markers in your file will remain untouched.

Slices each have their own play count, which is displayed in the bottom center of the slice. The play count will default to 1, but you can easily edit the count of an individual slice by double-clicking the number at the bottom of the slice and entering a value. To loop a slice infinitely, type `0, inf`, or in fact any text that is not a whole number.

To delete a selected slice, select it and hit delete on the keyboard, or drag its slice handle upwards out of the waveform view.

> ProTools users will find that markers from ProTools projects will not be included in bounced files, making QLab's automatic importing of makers somewhat less valuable. Fortunately, there is a workaround as long as you have a two-track editor that allows importing markers (such as TwistedWave):

1. Bounce or export your audio as usual.
2. In ProTools, choose *Export...* from the **File** menu and export your session info as text.
3. Open your audio file in your two-track editor and import the text file you created from ProTools. Et voila!

Integrated fade envelope

You can click the *fader* button to the right of the waveform to toggle the integrated fade envelope, which allows you to create a envelope that adjusts the overall volume over the course of the cue.

When the fade envelope is enabled, you can click and drag along the waveform to add fade points. Note that if you edit the envelope while the cue is playing, you won't hear your changes until you stop and restart the cue.

Rate and pitch

You can adjust the playback rate of the audio cue by typing a value in the *Rate* text field to the right of the waveform, or by clicking in that field and dragging up

or down. The *Pitch shift* option is checked by default, and will adjust the pitch correspondingly in a manner similar to changing the playback speed of tape.

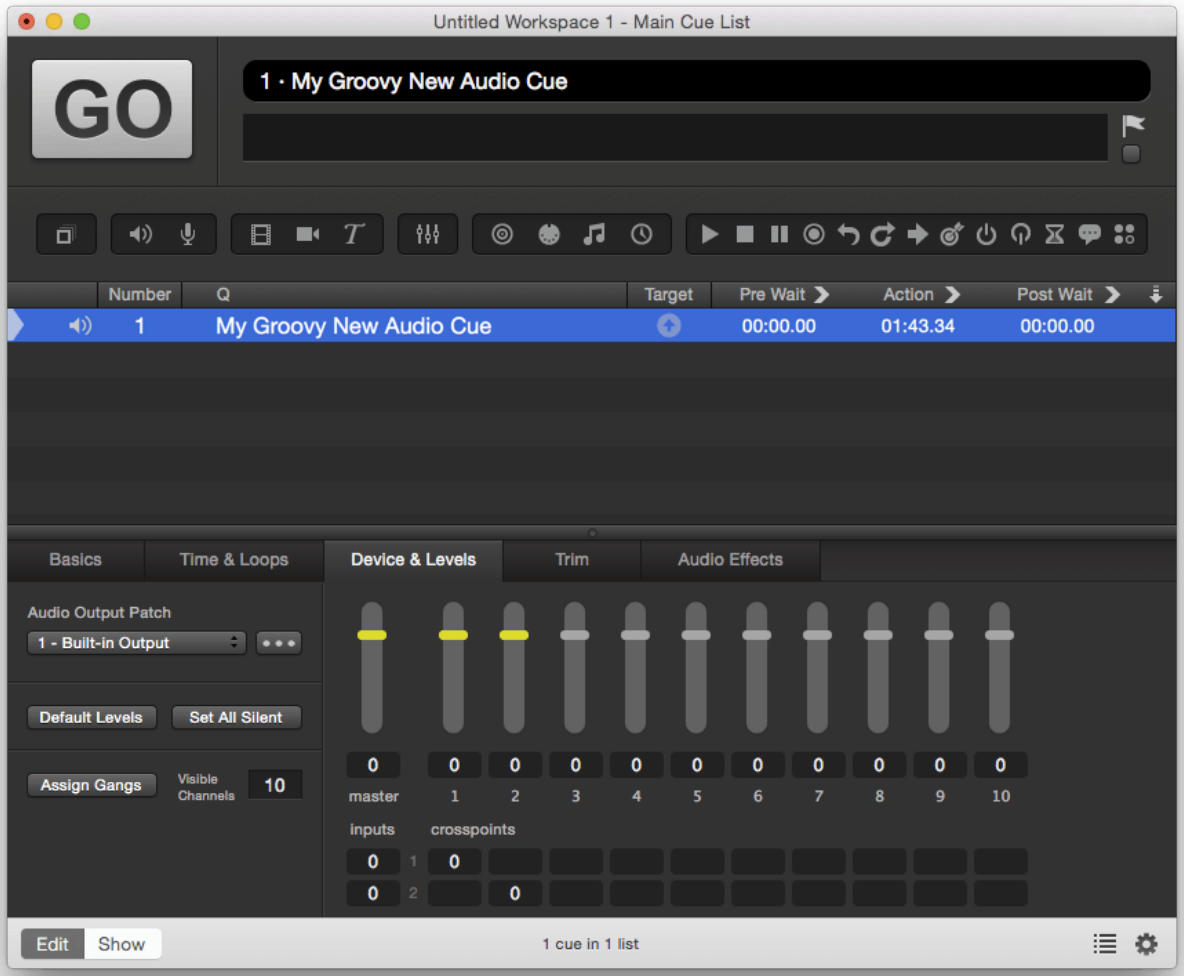
- Rate 1 = 100% (normal speed) - no pitch shift
- Rate .5 = 50% (half speed) - pitched down one octave
- Rate 2 = 200% (double speed) - pitched up on octave

The minimum rate is `0.03`, and the maximum rate is `33`.

To keep the pitch the same independent of the rate, simply uncheck the *Pitch shift* checkbox. It is worth noting that this is (perhaps counterintuitively) slightly more processor-intensive than leaving the option checked.

download video ↓

Device & Levels



Audio Output Patch. Choose your output patch from the drop-down menu in the top-left corner of the *Device and Levels* tab. The ••• button directly to its right a shortcut to the Patch Editor for the selected patch. Generally, you won't need to touch this after you set it up for your show. You can learn more about the Patch Editor in its section of this documentation.

Default Levels. This button sets all levels to the default levels, which can be set in the Audio section of Settings.

Set all silent. This button does exactly what you think it will.

Assign Gangs. When you click this button, all levels will become hidden; you then can type anything (for example, a single letter) into any or all of the levels fields. Any fields which you give the same text will become grouped together. When you click *Assign Gangs* again, ganged fields will appear with matching colored backgrounds so you can tell at a glance which fields are grouped. Then, you can simply click and drag one control to adjust every level in that group by the same amount.

If you gang levels that have different starting points, one level might reach its maximum or minimum level before another. Then and only then will the levels be adjusted disproportionately with one movement. Once the levels catch up to each other at their maximum or minimum, they will all move together.

Visible channels. This control allows you to set the number of output channels currently visible in the matrix mixer. Outputs which are not displayed are not disabled, they're simply hidden from view. This is only relevant if you have a Basic Audio, Pro Audio, or Pro Bundle license. The maximum number of outputs available with a Basic Audio license is eight. The maximum number of outputs available with a Pro Audio or Pro Bundle license is 48.

The Matrix Mixer

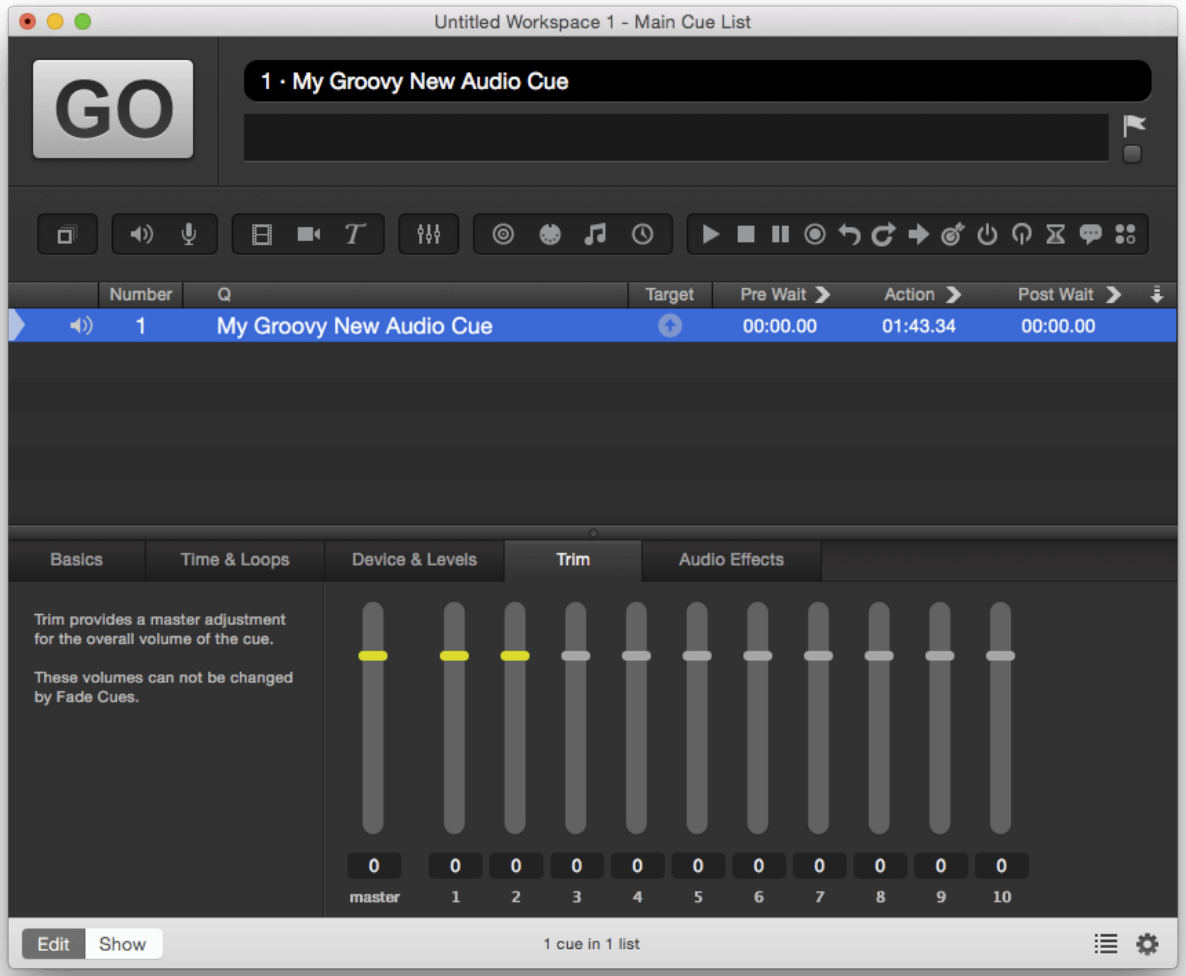
download video ↓

The matrix mixer is a grid made of rows and columns, like a spreadsheet. The master output level for the cue is the on the left, the cue outputs are the column headers, and channels in the audio file are the rows. Typically, there will be two rows: one for the left channel and one for the right, but QLab supports audio files of up to eight channels with a Basic Audio license or 24 channels with a Pro Audio or Pro Bundle license.

Crosspoints are the level controls for routing a given input (row) to a given output (column).

Levels can be typed in, or you can click and drag in any level field. QLab won't allow you to drag above `0 dB` as a safety measure, but you can type in any level above `0` manually (to the limit of whatever maximum level you set in audio settings).

Trim



The trim tab is similar to trim on a mixer: it provides a master adjustment for the overall volume of the cue. These volumes will not be affected by Fade Cues, but rather set an overriding master level for the cue.

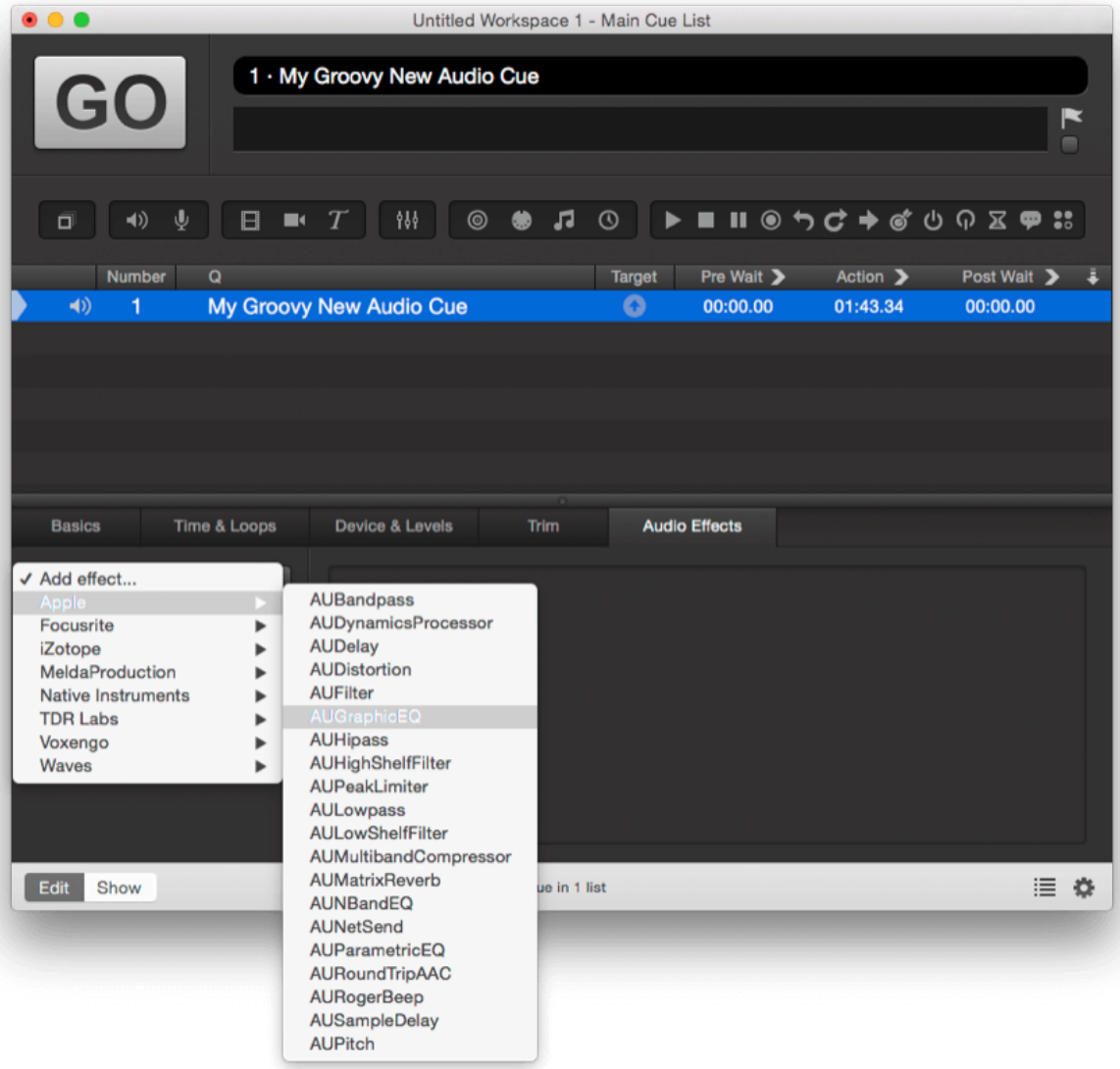
For example, if you make a cue sequence with a bunch of fades and lots of intricate level movement, and then afterwards you decide that one output is too loud throughout the whole sequence, you can adjust the trim in the audio cue instead of adjusting that output in each and every fade cue; just one adjustment and you're done.

Audio Effects

download video ↓

QLab gives you the ability to add audio effects to your cues, enabling you to process them dynamically from in realtime. In order to work in QLab, AudioUnits must be 64-bit, must be defined as an effect (i.e. MIDI synthesizers won't be available in QLab because they are generators, not effects), and must report a "tail time" (for example, a reverb's decay time.)

Additionally, the number of channels on your audio file needs to match the number of channels supported by the AudioUnit; if it does not, some AudioUnits will pass audio without rendering, and others won't pass any audio at all. A good example of this behavior is exhibited by Apple's *AUMatrixReverb*, which requires a two-channel (stereo) source. If you use AUMatrixReverb on a mono audio track, the audio will pass through the AudioUnit unchanged.



To apply an effect to an Audio cue, select it from the drop-down menu labeled *Add Effect*... When you select an effect, it will appear in the list to the right of the menu, and an AudioUnit editor window will open automatically. You may close the window and access it again easily anytime by clicking *Edit* next to the name of the effect in the list. The editor window looks different for each AudioUnit, as each effect requires different controls. QLab uses the built-in interface created by the designer of the AudioUnit, which means the look and feel (as well as quality and usability) of AudioUnits can vary widely.

Please note that meters in AudioUnits do not render at this time.

You can enable or disable effects by checking or unchecking the box to the left side of the effect, and you can delete effects by clicking the *X* to the right side. If the AudioUnit editor window is open, you can also turn the effect on or off by checking or unchecking *Enabled*.

Effects will be applied to the cue in the order in which they appear in the effects list. To change this order, simply click on an effect and drag it up or down within the list.

You can also insert AudioUnits on cue outputs and device outputs. You can find out more about using AudioUnits on outputs in the Patch Editor section of this documentation.

Slices and Vamping

download video ↓

Slices enable you to loop specific sections of an Audio cue or Video cue either a specific number of times, or an infinite number of times. If you set a loop in the middle of a cue, the Devamp cue gives you the ability to dynamically exit the loop and either continue to play through the cue, or trigger a following cue at the precise moment that the loop ends.

There are three ways to use a Devamp cue.

Devamp and Continue

First, create an Audio or Video cue with at least one slice towards the beginning of the cue, and set that slice to loop infinitely by double clicking the green slice count in the bottom of the waveform view, and typing `0` or `inf`. For the purposes of this example, it's a good idea for the slice to be longer than a few seconds.

Next, create a Devamp cue and target the Audio or Video cue with the looping slice. Look in the Settings tab of the inspector, and you'll see two checkboxes. Leave them unchecked.

Now, run the Audio or Video cue, and notice when it reaches the looping slice. It will keep repeating that slice indefinitely... until you run the Devamp cue. Run the Devamp, and then when the playback reaches the end of the slice, it will no longer loop, and instead proceed onwards.

If you have multiple looping slices in the Audio or Video cue, you can use multiple Devamp cues to pop out of each loop. Each Devamp cue will un-loop whichever slice is currently looping at the time the Devamp cue is triggered.

Devamp and Start Next

First, create an Audio or Video cue with at least one slice towards the beginning of the cue, and set that slice to loop infinitely by double clicking the green slice count in the bottom of the waveform view, and typing `0` or `inf`. For the purposes of this example, it's a good idea for the slice to be longer than a few seconds.

Next, create a Devamp cue and target the Audio or Video cue with the looping slice. Look in the Settings tab of the inspector, and you'll see two checkboxes. Check the box marked *Start next cue when target reaches the end of the current slice*.

Finally, create another Audio or Video cue directly after the Devamp cue.

Now, run the Audio or Video cue, and notice when it reaches the looping slice. It will keep repeating that slice indefinitely... until you run the Devamp cue. Run the Devamp, and then when the playback reaches the end of the slice, it will no longer loop, and instead proceed onwards.

At the same moment that the original cue passes the slice marker and continues onwards, the Devamp cue will trigger the following cue to play. This can be very useful for starting another, unrelated cue on a musical downbeat, or for starting a second layer of video in perfect time.

Devamp, Start Next, and Stop Target

First, create an Audio or Video cue with at least one slice towards the beginning of the cue, and set that slice to loop infinitely by double clicking the green slice count in the bottom of the waveform view, and typing `0` or `inf`. For the purposes of this example, it's a good idea for the slice to be longer than a few seconds.

Next, create a Devamp cue and target the Audio or Video cue with the looping slice. Look in the Settings tab of the inspector, and you'll see two checkboxes. Check the box marked *Start next cue when target reaches the end of the current slice*. and the box marked *Stop target when it reaches the end of the current slice*.

Finally, create another Audio or Video cue directly after the Devamp cue.

Now, run the Audio or Video cue, and notice when it reaches the looping slice. It will keep repeating that slice indefinitely... until you run the Devamp cue. Run the Devamp, and then when the playback reaches the end of the slice, it will stop.

At the same moment that the original cue stops, the Devamp cue will trigger the following cue to play. This can be useful for starting a new section of music after a downbeat or for putting a visual button on the end of a looping video.

Thinking In Bars and Beats

The Devamp cue enables QLab to behave as though it's aware of bars and beats by putting slice markers on each beat, and using Devamp cues to trigger actions that line up perfectly with those beats. As with all things, giving yourself plenty of time to experiment is the key to success.

Audio Signal Flow

Audio travels through QLab in this order:

1. File (for Audio Cues) or Audio Interface input (for Mic Cues)
2. Input controls
3. Cue Audio Effects
4. Crosspoint controls
5. Cue Output controls
6. Master Level control
7. Cue Output Audio Effects
8. Device Routing matrix
9. Device Output Audio Effects
10. Audio interface control software (if applicable)
11. Audio interface

Broken Cues

Audio cues can become broken for the following reasons:

Invalid Audio File

Either the file is missing or damaged, or it's not one of the supported audio file types.

No audio device. Select one in the "Device & Levels" tab.

You may also need to visit the Audio section of Settings and connect an audio device to the desired patch.

One or more audio effects in this cue are missing.

This typically happens when a workspace that uses audio effects is moved from one computer to another, and the new computer does not have the necessary AudioUnit installed. Either choose a different audio effect, or install the necessary AudioUnit.

One or more audio effects on the cue outputs are missing.

This typically happens when a workspace that uses audio effects is moved from one computer to another, and the new computer does not have the necessary AudioUnit installed. Either choose a different audio effect, or install the necessary AudioUnit.

One or more audio effects on the device outputs are missing.

This typically happens when a workspace that uses audio effects is moved from one computer to another, and the new computer does not have the necessary AudioUnit installed. Either choose a different audio effect, or install the necessary AudioUnit.

A pro license is required to use audio effects.

You'll need to either install a Pro Audio or Pro Bundle license, or remove the audio effects from this cue.

Timecode triggers require a Pro license.

You'll need to either install a Pro Audio or Pro Bundle license, or remove the timecode trigger from this cue.

Mic Cues

Mic cues are similar to Audio cues, except instead of targeting a track on your computer, Mic cues take in live audio from a microphone or any live source of audio plugged into your audio interface.

When a Mic cue is selected, four tabs will appear in the Inspector:

- Basics
- Device & Levels
- Trim
- Audio Effects

Basics

Please refer to the section on the inspector in the "Getting Started" section of this documentation.

Device & Levels

Audio input & output patch. A Mic cue must use the same audio device for both its input and its output. This can be the same device used for Audio cues in the same workspace, but it must be set up separately and will appear as a separate device in QLab. (See Settings.) If you need to use one device for input and a separate device for output, you must set up an aggregate audio device (using an

Apple-supplied application called Audio MIDI Setup, found in Applications/Utilities. The process is described in this Apple support document.)

More The ••• button is a shortcut to open the patch editor for the selected patch. The patch editor lets you name cue outputs, assign audio effects to cue outputs, route cue outputs to device outputs, and assign audio effects to device outputs. For more information about this, please see the Audio section of the Settings page of this documentation.

Default Levels. Clicking this button sets the levels of the cue to the default levels, which can be set in the Mic section of Settings.

Set all silent. This button does exactly what you think it will.

Assign Gangs. Assigning gangs for Mic cues is exactly the same as for Audio cues; please refer to Audio Cues.

Visible Channels. This control allows you to set the number of output channels currently visible in the matrix mixer. Outputs which are not displayed are not disabled, they're simply hidden from view.

Matrix Mixer. In an Audio cue, each row in the Matrix Mixer represents one channel of the targeted audio file. In a Mic cue, each row represents one input on your audio device. QLab will automatically make a row for every input it recognizes. If, for example, you're using the Mac's built-in input you'll see two rows, but if you're using an RME FireFace 800 you will see 24 rows, because the FireFace 800 has 24 inputs (actually it has 28 inputs, but QLab only supports 24 inputs for Mic cues). The rest of the matrix mixer works the same as in Audio Cues.

download video ↓

Trim

Trim for Mic cues operates the same as Trim for Audio Cues.

Effects

Effects for Mic cues operate the same as Effects for Audio Cues.

Broken Cues

Mic cues can become broken for the following reasons:

No audio device. Select one in the “Device & Levels” tab.

You may also need to visit the Mic section of Settings and connect an audio device to the desired patch.

One or more audio effects in this cue are missing.

This typically happens when a workspace that uses audio effects is moved from one computer to another, and the new computer does not have the necessary AudioUnit installed. Either choose a different audio effect, or install the necessary AudioUnit.

One or more audio effects on the cue outputs are missing.

This typically happens when a workspace that uses audio effects is moved from one computer to another, and the new computer does not have the necessary AudioUnit installed. Either choose a different audio effect, or install the necessary AudioUnit.

One or more audio effects on the device outputs are missing.

This typically happens when a workspace that uses audio effects is moved from one computer to another, and the new computer does not have the necessary AudioUnit installed. Either choose a different audio effect, or install the necessary AudioUnit.

A pro license is required to use audio effects.

You’ll need to either install a Pro Audio or Pro Bundle license, or remove the audio effects from this cue.

Timecode triggers require a Pro license.

You'll need to either install a Pro Audio or Pro Bundle license, or remove the timecode trigger from this cue.

Fading Audio

A Fade cue can be used to adjust the volume levels and audio effect parameters of a targeted Audio or Mic cue. Fade cues can also target Video cues, Camera cues, and Titles cues; when a Fade cue is selected, the inspector will only show the tabs relevant to the type of cue that the Fade cue is targeting.

Fade cues require a target and a duration, and must adjust at least one level or audio effect parameter.

To learn how to set a target for a Fade cue, please refer to the section on targeting other cues in the **Getting Started** section of this documentation.

When a Fade cue which targets an Audio or Mic cue is selected, four tabs appear in the Inspector:

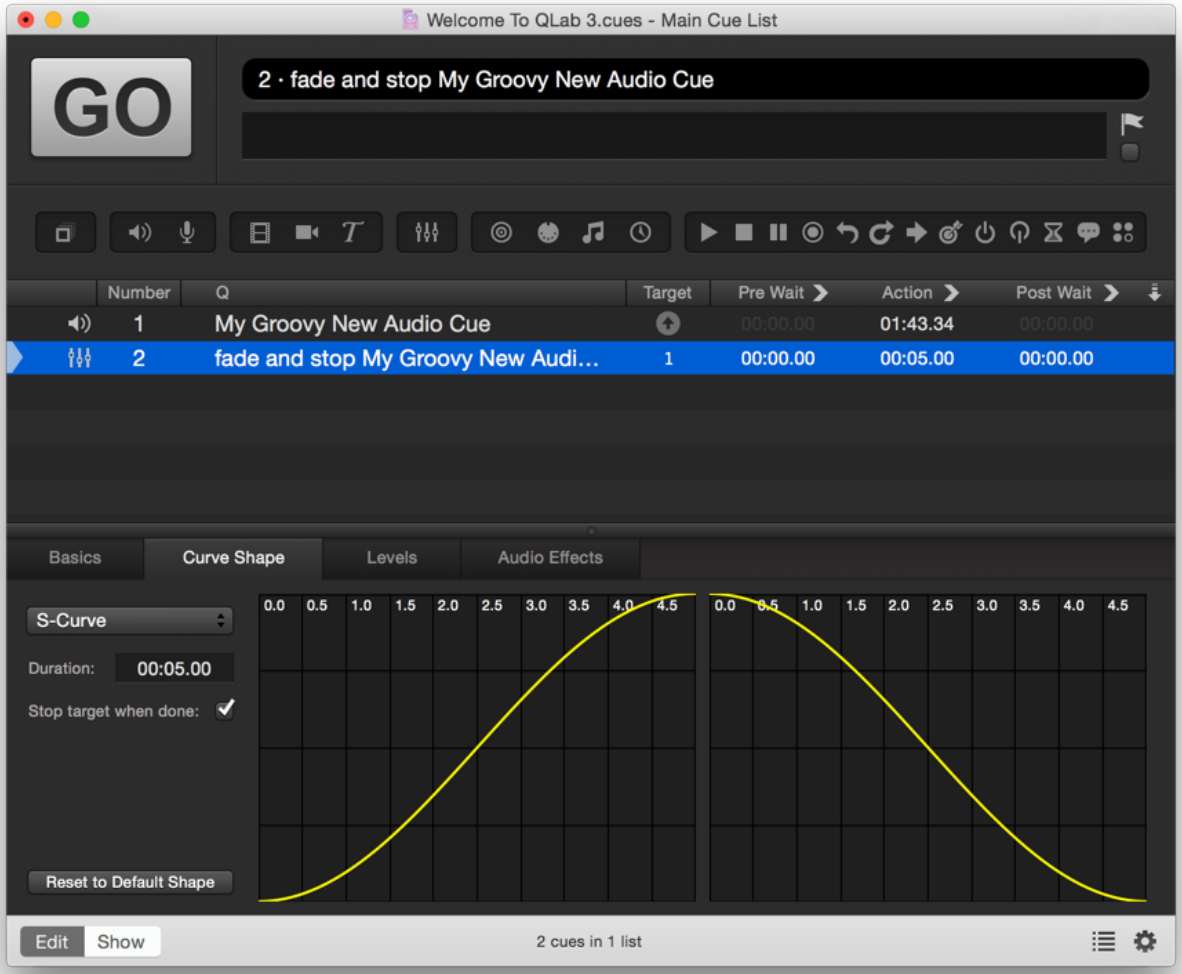
- Basics
- Curve Shape
- Levels
- Audio Effects

Basics

Please refer to the section on the inspector in the **Getting Started** section of this documentation.

Curve Shape

The curve shape determines the manner in which the parameter or parameters are adjusted over the course of the fade. QLab defaults to an S-curve, but any curve can be drawn in the Curve Shape tab by selecting *Custom Curve* from the drop-down menu in the top left corner of the tab. The curve is drawn in yellow on the right side of the tab. The curve on the left is the shape for levels which are increasing, and the curve on the right is for levels which are decreasing. Click anywhere along the yellow line and a yellow dot, called a control point, will appear. Drag it to change the shape of the curve. Clicking on the curve again to create more control points. The active point will be filled-in yellow circle, and others are yellow outlines. To delete one, click on it to select it and press the **delete** key on your keyboard. To start over entirely, click *Reset to Default Shape* in the bottom left corner of the tab.

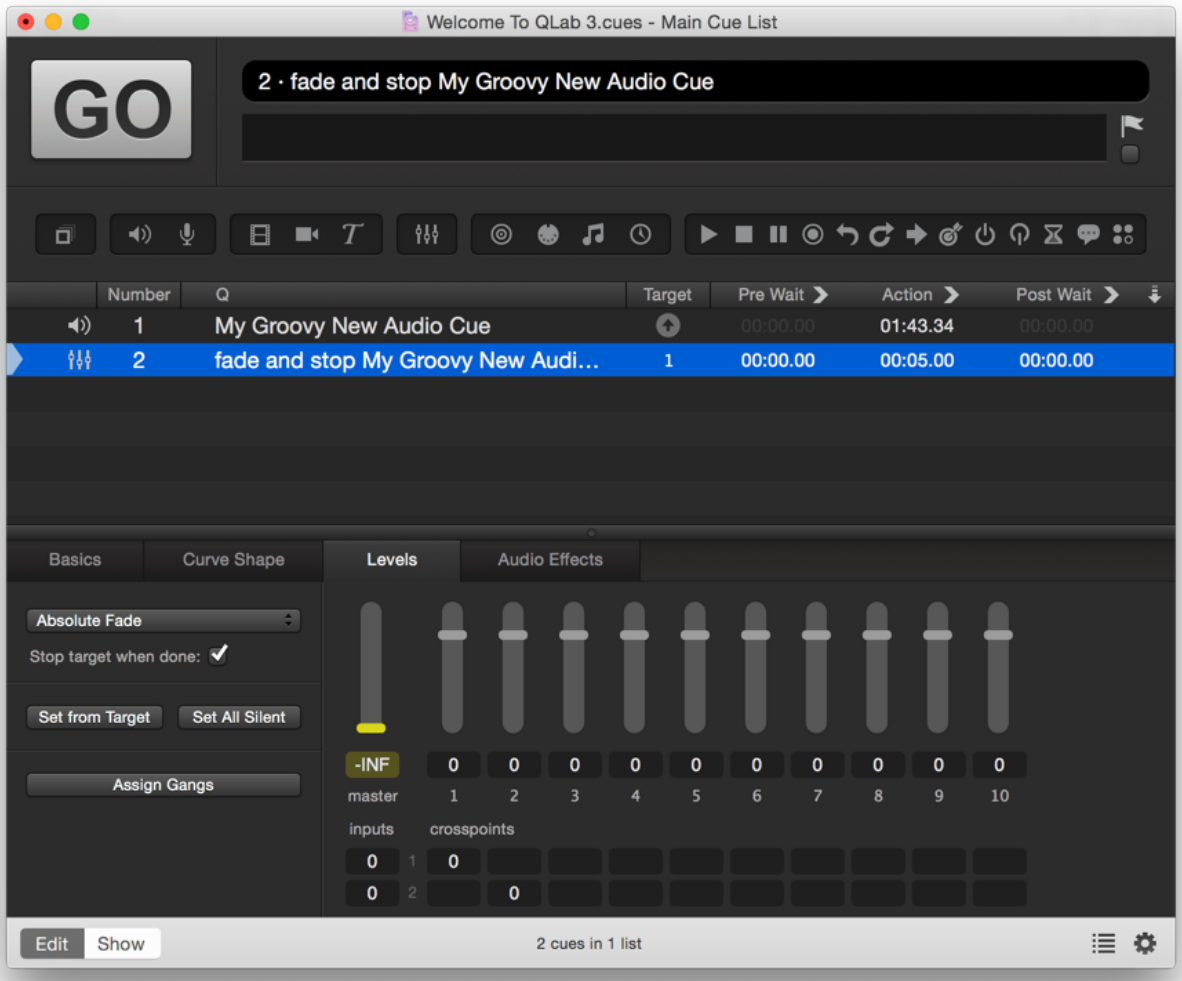


Check or uncheck the *Stop target when done* box under the Duration text field depending on whether you would like the target cue to continue playing after the Fade cue is complete, or stop once the Fade is complete.

Levels

The Levels tab allows you to specify which audio levels you wish to fade, and what their final volume will be. You can specify a different volume for each level.

The right side of the Levels tab looks and behaves like the Matrix Mixer, and will only display as many input channels as exist in the target cue. Only “active” controls will actually be applied to the target cue when the fade cue is fired; this allows you to choose which channels of the sound you want to fade. You can activate or deactivate a control by clicking on it. It will turn yellow when it is active, and grey when it is not. Changing the volume of any slider or knob will automatically activate it.



(See that yellow slider there? That control is active. The rest are grey, and thus inactive.)

The volume of an audio level is specified in decibels (dBFS). The default range of volumes in QLab is `+12 dB` to `-60 dB`, which is treated by QLab as `-∞`, displayed as `-INF`. Those limits can be changed in Settings.

Click the *Gangs* button on the left side of the Levels tab to assign gangs to any of the level controls. Each channel in the same gang will be adjusted together when you change the volume of one member of the gang.

Click the *Set from target* button on the left side of the Levels tab, or use the keyboard shortcut ⇧⌘**T**, to snap all levels to the levels of the target cue, but keep them deactivated. This is a convenient way to keep track of the levels from which you will be fading.

Click the *Set All Silent* button on the left side of the Levels tab to return the fade to a pristine state.

There is a *Stop Target When Done* button in the Levels tab as well as in the Curve Shape tab. They are not different.

Audio Effects

download video ↓

When you apply an effect to an Audio cue, any Fade cues that target that Audio cue will automatically recognize the effects you've applied, and they will be listed in the Audio Effects tab of the inspector for the Fade cue. By default, the checkbox next to an effect will be unchecked, meaning that the Fade cue will not adjust parameters of that effect. To control an effect with the Fade cue, just check box next to the effect.

Unlike fading audio levels, there is no way to activate or deactivate individual parameters of an audio effect in a fade. Thus, to avoid accidentally adjusting more parameters than you intend, you can click *Set Audio Effects from Target* to copy the levels of every parameter in the effect from the target cue, and paste them into the Fade cue. Otherwise, the levels in the Fade cue will default to the built-in default levels for that audio effect.

Once you've done that, or elected not to, you can click the *Edit* button for the effect to adjust the effect.

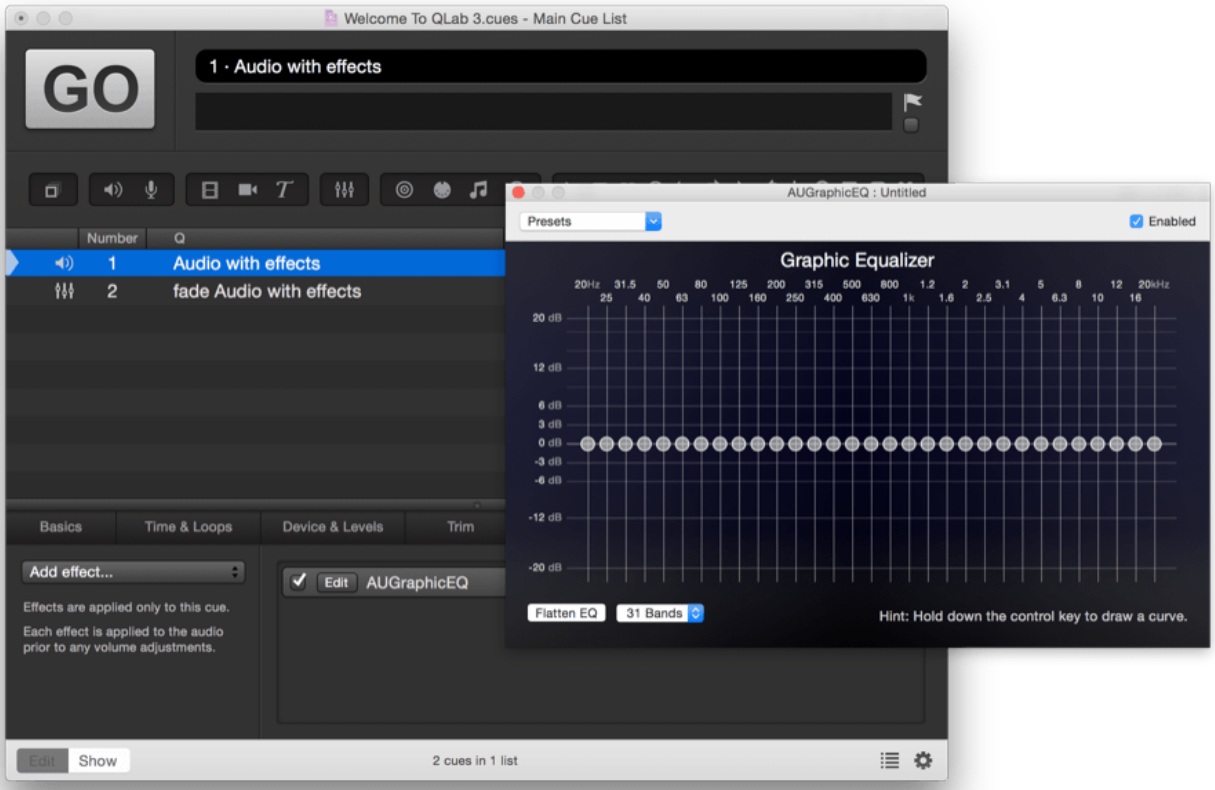
It's important to understand that when you click *Edit* to view the effect, you're not opening the same window as when you click *Edit* in the Audio cue. It can be difficult to keep track of which edit window belongs to which cue. Take your time and be sure that you're making changes in the place you intend.

Make any adjustments you wish to the effect. This is a situation in which *Live fade preview* can be very helpful. It's turned on by default, but if you've turned it off, consider turning it back on when adjusting audio effects in a Fade cue. That way, you can make adjustments in the Fade cue while the target Audio cue is running, so that you can hear those adjustments in real time.

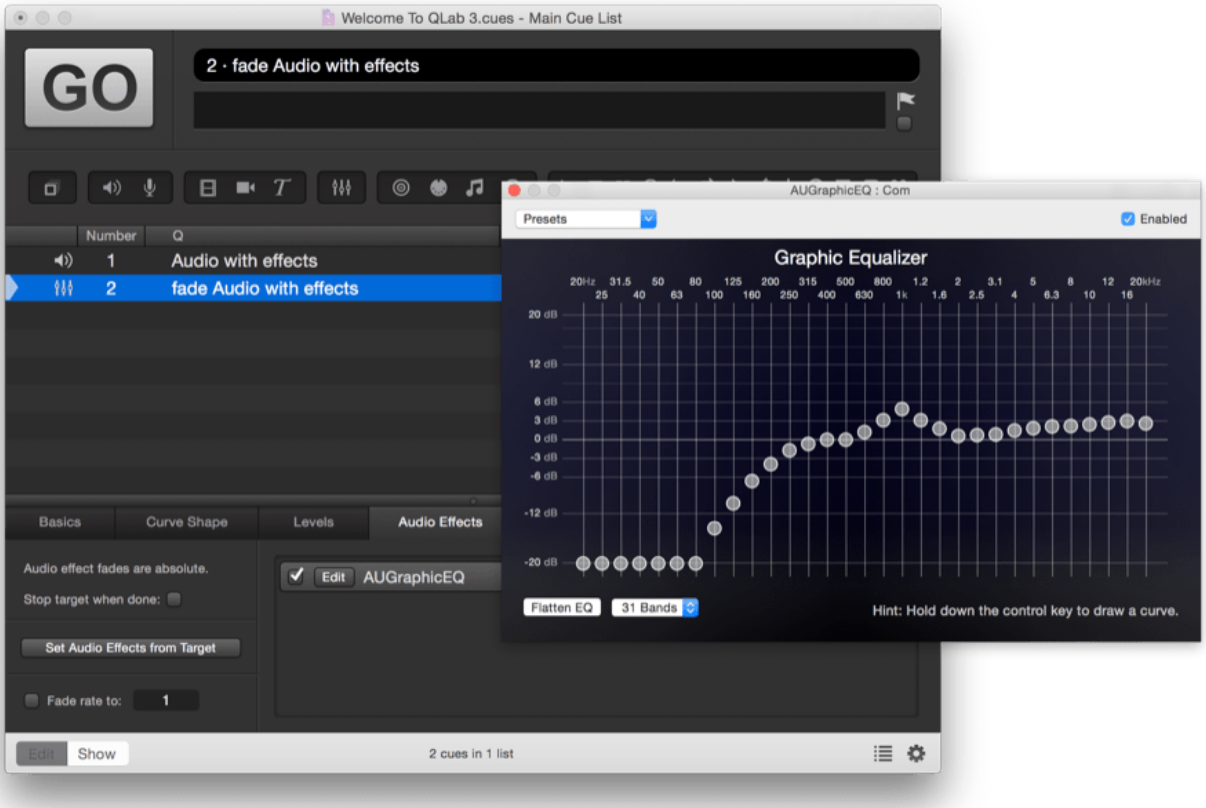
When you're done, close the editor window.

Now, when you run the Fade cue, the parameters you adjusted will smoothly fade from their initial values, set in the Audio cue, to the values you set in the Fade cue.

If you're new to effects editing, try starting with the AUGraphicEQ effect. Create a new Audio cue with a piece of music, and add the AUGraphicEQ effect.



Then, create a Fade cue, target the Audio cue, click “Set Audio Effects from Target” and then open the effect editor for the Fade cue. Make a dramatic adjustment. Play the Audio cue, then the Fade cue.

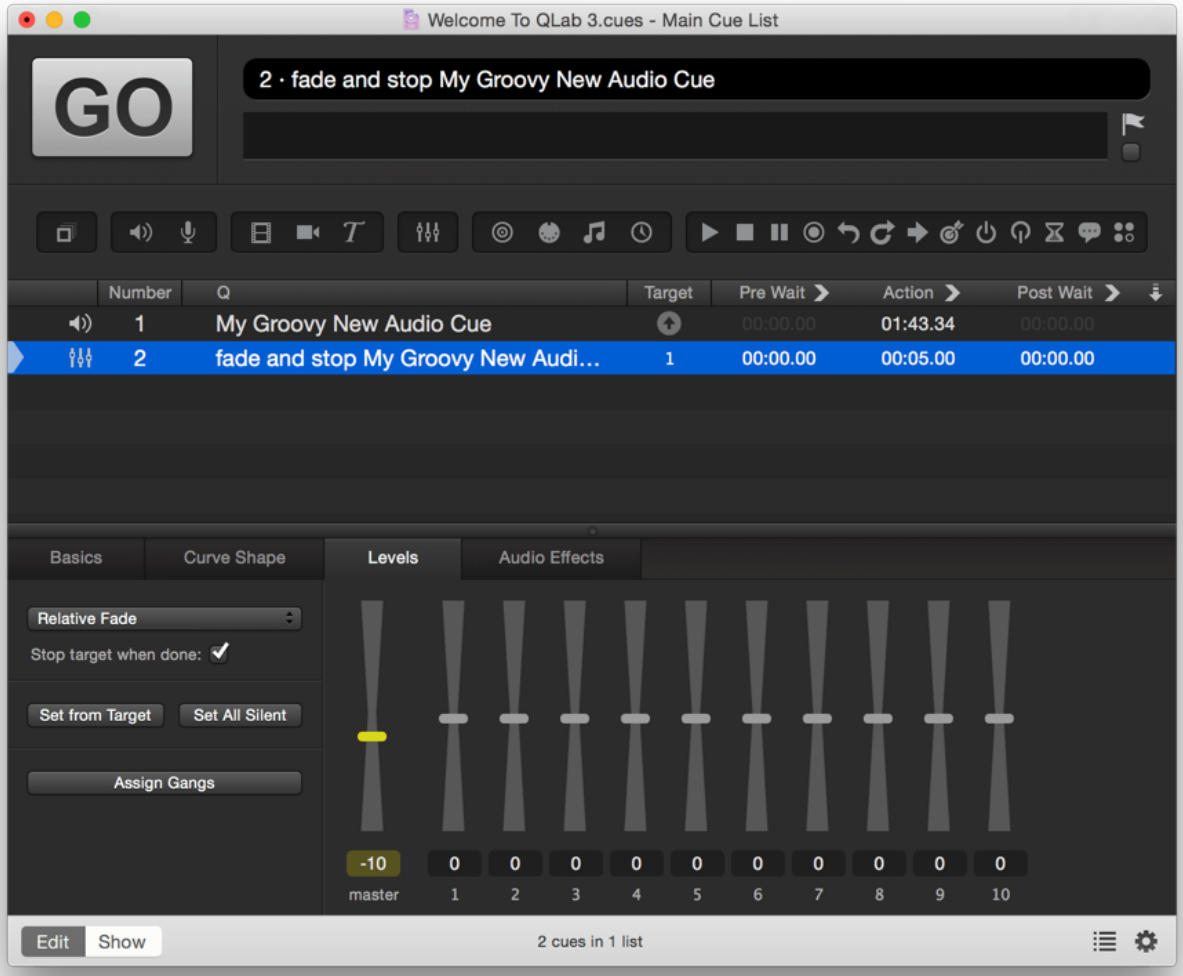


AudioUnits can also be placed on Cue Outputs and on Device Outputs, but effects there cannot be the target of Fade Cues. To learn more about effects on outputs, see Settings.

Relative Audio Fades

By default, Fade cues are *Absolute*. This means that any parameters that you adjust with a Fade cue will arrive at their final levels regardless of their status before the Fade cue runs. If you use a Fade cue to set the master level of a target Audio cue to `-12`, then the master level of that Audio cue will end up at `-12` after running the Fade no matter what the level was when the Audio cue started off.

However, using the drop-down menu at the top left corner of the Levels tab, you can set a Fade cue to be a *relative* fade. Relative fades add or subtract a given amount to the active parameters, so the starting point of those parameters very much matters.



In this screen shot, the Fade cue reduces the master level of the target cue by 10 db. If you run the Fade cue twice, the master level of the target cue will end up `-20` below its starting level.

Be very careful with relative fades that raise levels. If, for example, you were to create a relative Fade cue to raise the level of an Audio cue from a very soft level,

say `-50`, up to `-10`. You'd set the Audio cue to `-50`, then create a relative Fade cue with a level change of `+40`. If you accidentally ran that Fade cue twice, QLab would attempt to fade the Audio cue to `+30`, which is rather loud!

To help protect you against this, QLab has a the maximum level set in the Audio section of Settings. By default, this is `+12`, meaning that in the example above, the Audio cue would end up at `+12`, not `+30`. Depending on the gain structure of your sound system, though, this might still be uncomfortably loud.

Routing and Panning

While fading in, fading out, and adjusting the overall volume of an Audio cue are probably the most common uses of a Fade cue, you can also use a Fade cue to pan an Audio cue amongst any pair or group of outputs.

The best way to understand this is to walk through a simple example. Let's say you have an Audio cue that targets a mono track, and you want to pan it from one speaker to another. QLab will recognize the track has one channel, and one row will appear in the *Device & Levels* tab on the inspector for that Audio cue. In the Audio cue, set the level for the first speaker (cue output 1, the first column of the matrix) to `0`, and set the level for the second speaker (cue output 2, the second column of the matrix) to `-INF`. (Note that QLab treats a blank field in this matrix as `-INF`.)

Next, create a Fade cue which targets the Audio cue, and reverse the levels: set the first output to `-INF` and the second output to `0`. When you play the Audio cue, you will hear it exclusively from output 1. When you play the Fade cue, it will fade down in output 1 and fade up in output 2 over the same duration.

One fade cue can affect as many controls as you wish. If instead of a mono track you have a track with eight channels, and instead of two speakers you have ten,

you can enter values in the matrix or adjust the sliders to have different channels come up or down in different outputs.

A key concept here is that if multiple Fade cues share a target, but each Fade cue has different active controls, then those Fade cues can run simultaneously without interfering with each other. Because of this, in the scenario with an eight-channel Audio cue fading amongst ten speakers, very complex fades can be achieved by using simultaneous Fade cues with different active controls, different curve shapes, and different durations.

Reverting Fades

QLab has a sort of special-case *undo* command that applies only to Fade cues, called *Revert Fade Action*. You can find this command under the Tools menu when a Fade cue is selected, or you can use the keyboard shortcut ⇧⌘**R**.

When *Revery Fade Action* is invoked on a Fade cue after that Fade cue has been run, QLab reverts the levels of the target of the selected Fade cue to whatever they were before the Fade ran *except* for levels which have been otherwise changed. That is to say, only the adjustments that the selected Fade cue caused are reverted.

Broken Cues

Fade cues can become broken for the following reasons:

No target cue.

Assign a target to the cue.

No fade parameters have been enabled; pick at least one thing to fade.

You can enable or disable an audio control by clicking in it; active controls will be highlighted in yellow. You can enable or disable a video control by checking or unchecking the box next to it.

One or more audio effects in this cue are missing or broken.

This typically happens when a workspace that uses audio effects is moved from one computer to another, and the new computer does not have the necessary AudioUnit installed. Either choose a different audio effect, or install the necessary AudioUnit.

A basic or pro video license is required to fade video geometry.

You'll need to either install a Basic Video, Pro Video, or Pro Bundle license, or adjust the Fade cue to not fade video geometry parameters.

A pro license is required to fade the playback rate.

You'll need to either install a Pro Audio, Pro Video, or Pro Bundle license, or adjust the Fade cue to not fade rate.

A pro audio license is required to fade audio effects.

You'll need to either install a Pro Audio, Pro Video, or Pro Bundle license, or adjust the Fade cue to not fade audio effects.

Timecode triggers require a Pro license.

You'll need to either install a Pro Audio or Pro Bundle license, or remove the timecode trigger from this cue.

Patch Editor

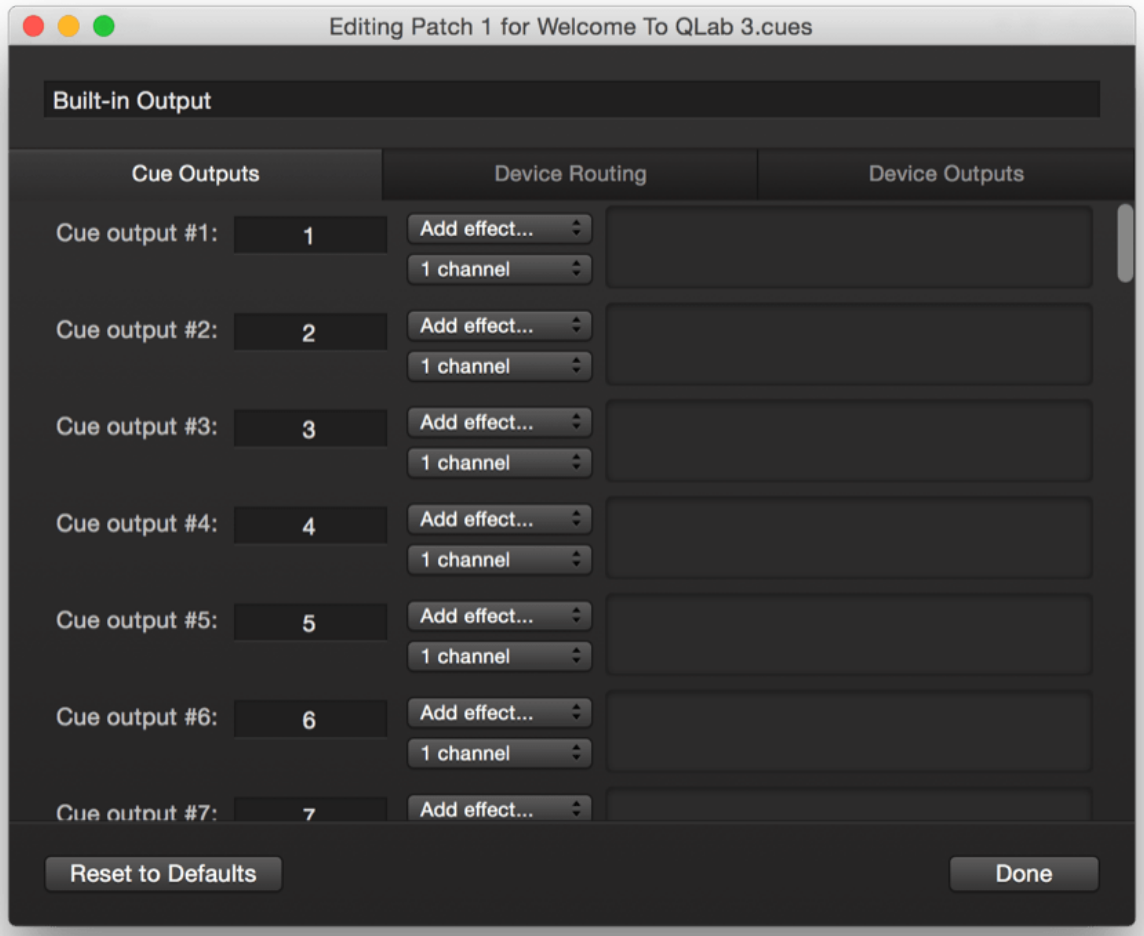
The Patch Editor allows you to edit the routing of Cue Outputs to Device Outputs, and to add AudioUnit effects to either.

The field at the top of the Patch Editor lets you change the name by which QLab identifies the audio device. Changing the name is purely for convenience, and has no effect outside of QLab.

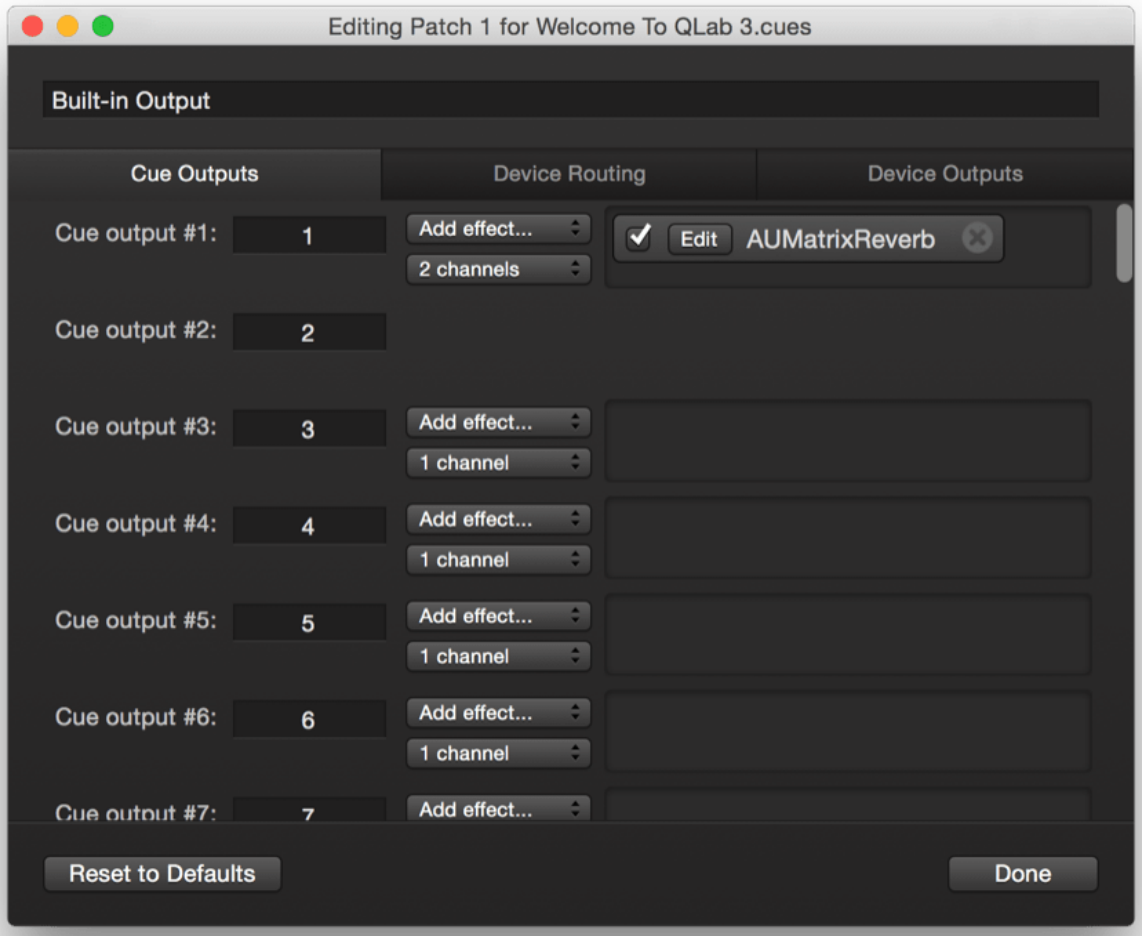
Cue Outputs

Cue outputs can be thought of as busses or mixes on a traditional audio console. Sound from a cue routes to cue outputs, which are in turn routed to device outputs, which are QLab's representation of the actual physical outputs on your audio interface. With a Pro Audio or Pro Bundle license installed, QLab has 48 cue outputs. A Basic Audio license gives you eight cue outputs. In any other case, QLab gives you two cue outputs.

You can rename cue outputs from their default numbers to any text you wish. The changed names will be reflected in the Device & Levels tab of the inspector.



The *Add effect*... dropdown menu will show you a list of all 64-bit compatible AudioUnits installed on the Mac, and by choosing from this list you can apply an AudioUnit to the cue output, much like applying an effect to an aux send on a console or in a DAW. Some AudioUnits (for example, Apple's AUMatrixReverb) require stereo input, and so you can use the *1 channel* dropdown menu to link a cue output to the following cue output, and use them as a stereo pair for the purposes of driving stereo effects.

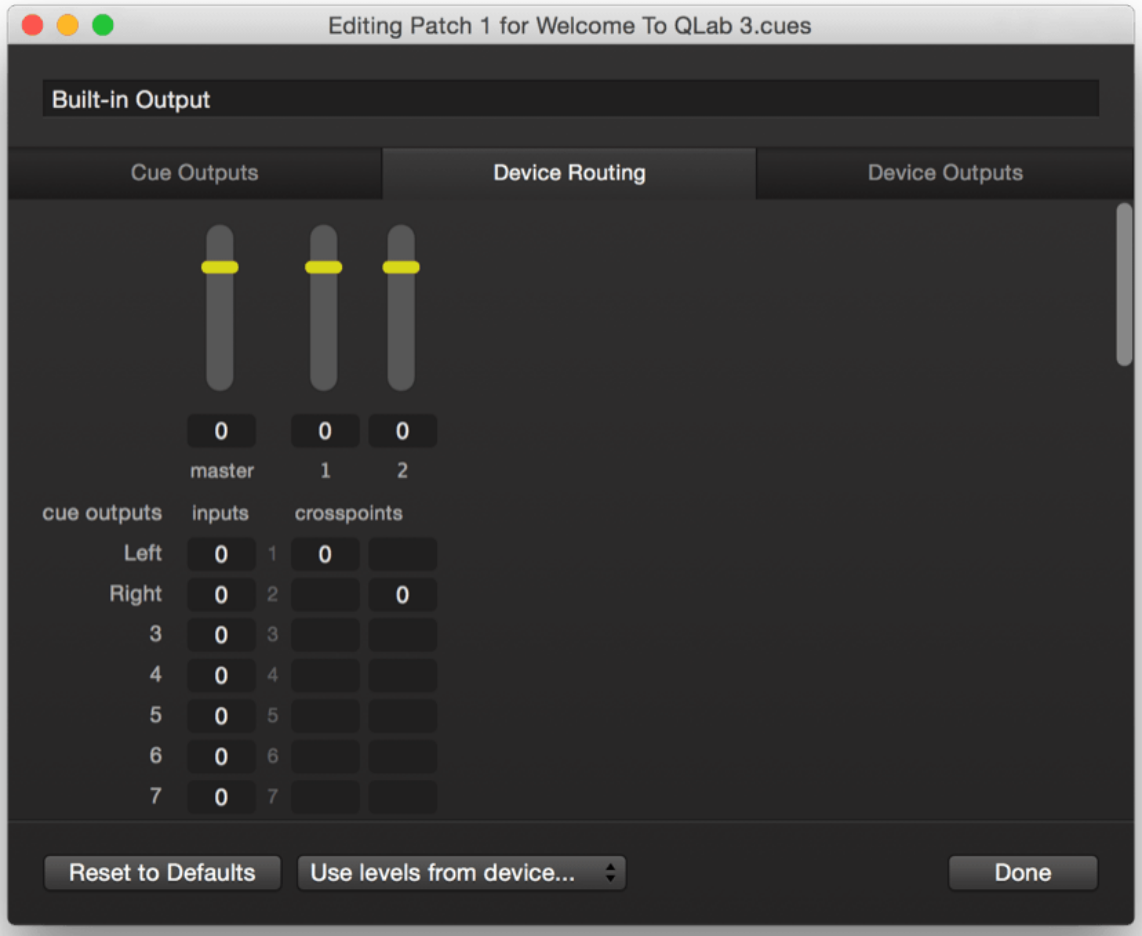


You can insert multiple AudioUnits on each cue output, and drag them left to right to set the order in which they are inserted.

It's important to note that running AudioUnits is much more processing intensive than just playing straight audio. It's not difficult to bring even a powerful Mac to its knees by loading up a bunch of effects, particularly more complex ones such as convolution reverbs.

Device Routing

In this matrix mixer, rows represent cue outputs, and columns represent device outputs. The default routing is one to one, but you can route any or all cue outputs to any or all device outputs at any level.

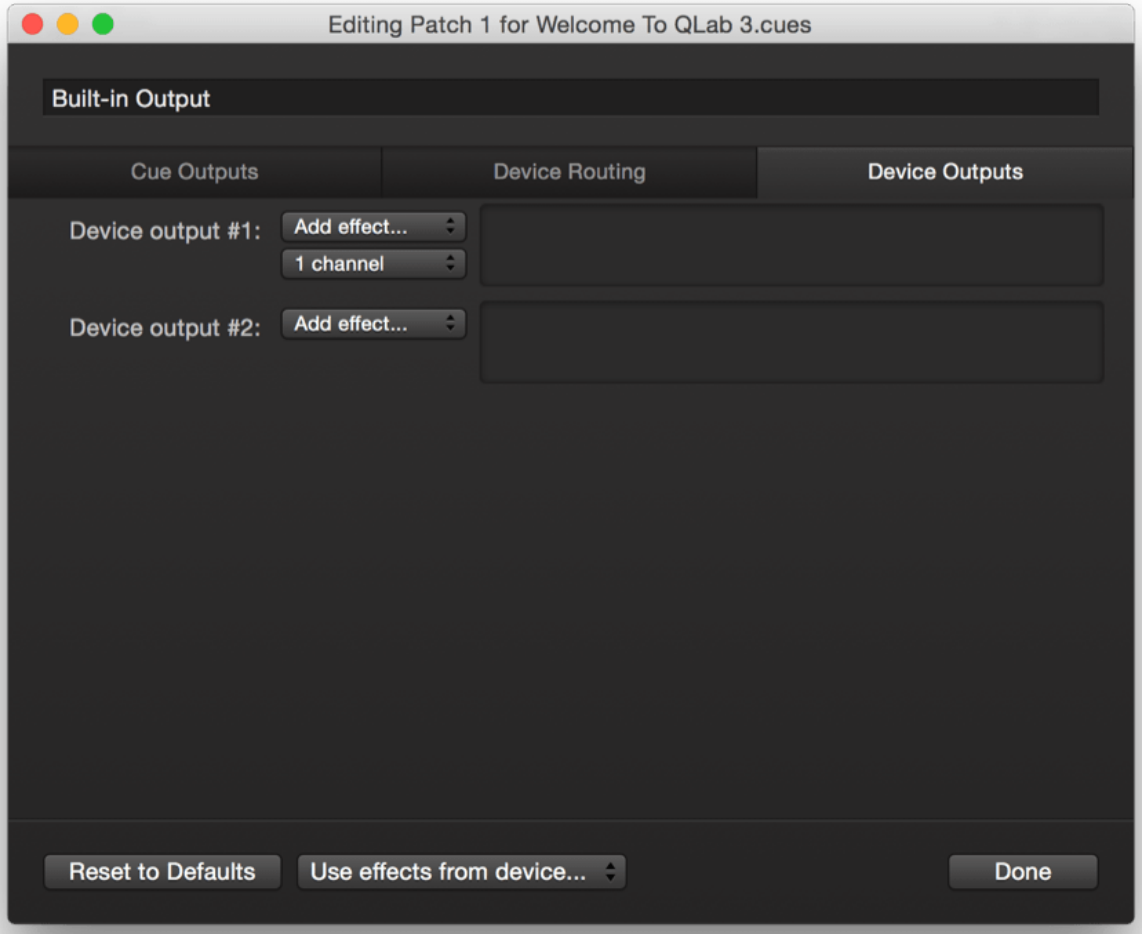


The Reset to Defaults button at the bottom restores the matrix to its default position.

The *Use levels from device...* button shows a list of any audio devices that have ever been attached to this workspace, and selecting one attempts to set the current routing to match that of the device when it was last connected.

Device Outputs

Here, you can add AudioUnit effects to the very end of the signal path within QLab. Just like with cue outputs, you can connect device outputs into pairs in order to drive stereo-only AudioUnits.



The *Use effects from device...* button shows a list of any audio devices that have ever been attached to this workspace, and selecting one attempts to duplicate the arrangement of AudioUnits used with that device when it was last connected.

Audio Signal Flow

Audio travels through QLab in this order:

1. File (for Audio Cues) or Audio Interface input (for Mic Cues)
2. Input controls
3. Cue Audio Effects
4. Crosspoint controls
5. Cue Output controls
6. Master Level control
7. Cue Output Audio Effects
8. Device Routing matrix
9. Device Output Audio Effects
10. Audio interface control software (if applicable)
11. Audio interface

Chapter 4: Video

Video Cues

download video ↓

Video cues require a target, which must be a file containing video. While QLab can play files in any format supported by AVFoundation, we recommend the following formats, listed in order of preference, for videos without transparency:

- PRORES 422 PROXY
- PRORES 422 LT
- PHOTOJPG
- H.264

For videos with transparency, also referred to as alpha channel support, AVFoundation only supports PRORES 4444.

Video cues can also play still images in all common formats. We recommend PNG and JPG.

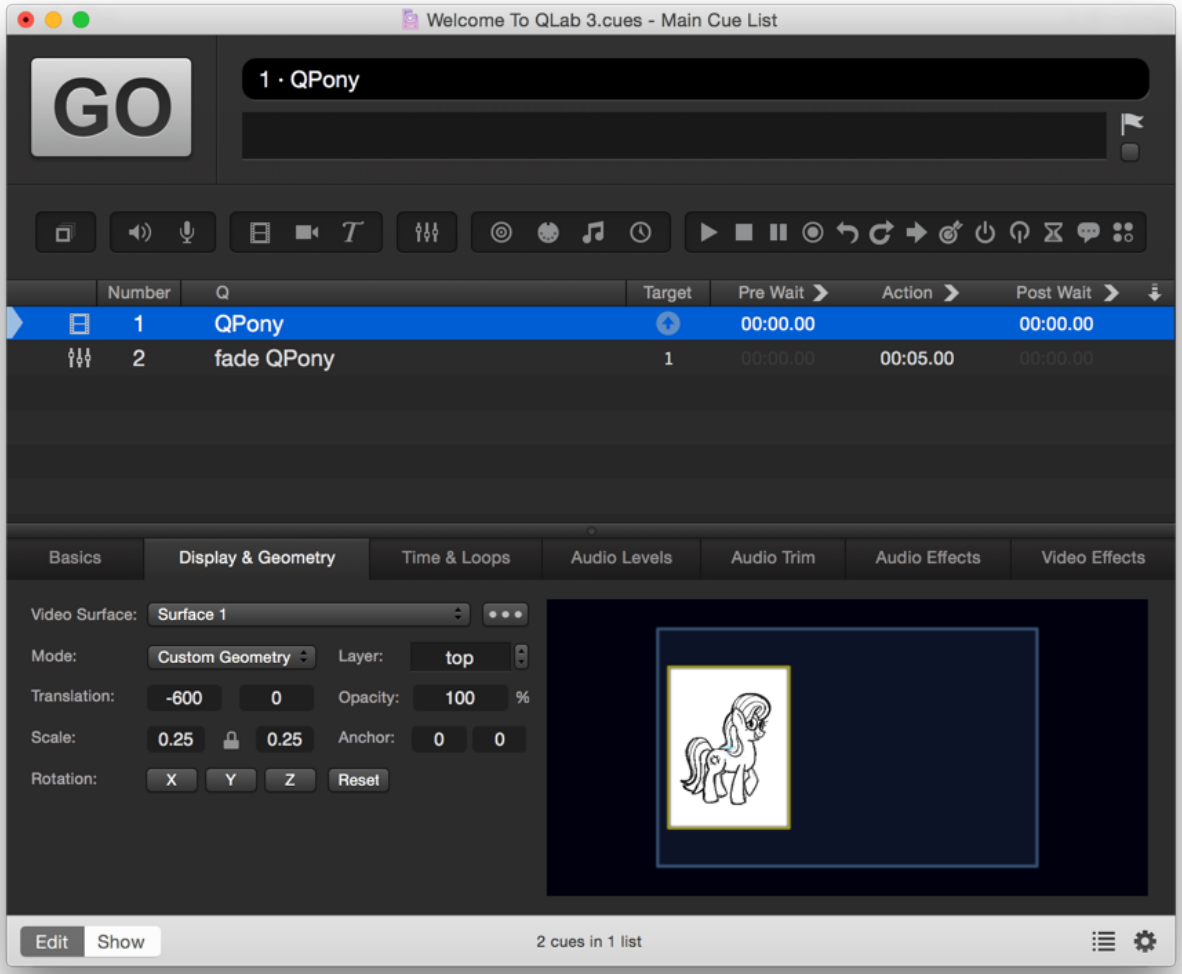
When a Video cue is selected, seven tabs will appear in the Inspector:

- Basics
- Displays & Geometry
- Time & Loops
- Audio Levels
- Audio Trim
- Audio Effects
- Video Effects

Basics

Please refer to the section on the inspector in the **Getting Started** section of this documentation.

Display & Geometry



Video Surface. Choose on which screen you would like the Video cue to play from the drop-down menu.

Mode. Select either *Full Screen* or *Custom Geometry* from the drop-down menu.

Layer. QLab handles layers differently from most media servers. Rather than thinking of layers as a container into which individual cues get placed, think of layers as the order in which video cues stack up on top of each other. Therefore, multiple cues can be assigned to the same layer at the same time.

QLab has 1001 layers: top is closest to the viewer, then 999, then downwards to 1, then bottom. Any cues assigned to the top layer will play on top of all other currently running cues. Any cues assigned to the bottom layer will play beneath all other currently running cues. Cues assigned to the same layer as each other will stack in the order in which they are triggered. So, for example, if you have three cues assigned to layer 10, whichever cue is triggered last will appear on top of the other two, but any cue assigned to layer 11 will appear on top of all three cues on layer 10, regardless of the order in which those cues are triggered.

Video cues will play on the top layer by default. You can change layers by typing in a value, or with the arrow keys.

Opacity. Enter values from 0% to 100% to adjust the opacity of the cue.

Full Screen. When you select this option from the Mode menu, there will be a checkbox labeled *Preserve Aspect Ratio*. With this box checked, the cue will play as large as possible without distorting in either axis. With the box unchecked, the cue will stretch to fill the Surface to which it is assigned.

Custom Geometry. When you select this option from the Mode menu, more controls become available to you:

Translation. Enter values to move the video along the X and Y axes, or click and drag the video within the preview thumbnail to the right of the controls to move it.

Scale. Enter values to enlarge or shrink the cue. If the lock icon is closed, the aspect ratio will be preserved. Click the lock to open it and adjust the height and width independently.

Anchor. Specify an anchor, which is the point around which a video cue rotates. The anchor is also the center point of the video for the purposes of zooming in and out. You may enter numerical values along the X and Y axis, or click and drag

the light blue cross in the preview thumbnail to the right of the controls to place the anchor.

Rotation. Click X, Y, or Z to manipulate rotation around the desired axis and a text field will appear. Enter a value for the desired angle of rotation. Start over by clicking the *Reset* button.

A Word About Quaternions

QLab uses quaternion math to handle rotation of Video cues in 3D space. The advantage to quaternions is that when fading a Video from one position to another, QLab will always produce smooth, natural motion. The tradeoff is that there is no useful way to numerically display the current rotation of a cue. Therefore, when you adjust the rotation of a cue, you'll see numbers in the pop-up which represent the degrees of single-axis rotation since you started this particular adjustment, not any sort of absolute measure.

Time & Loops

Time & Loops for Video cues operate the same as Time & Loops for Audio cues.

Audio Levels, Audio Trim, and Audio Effects

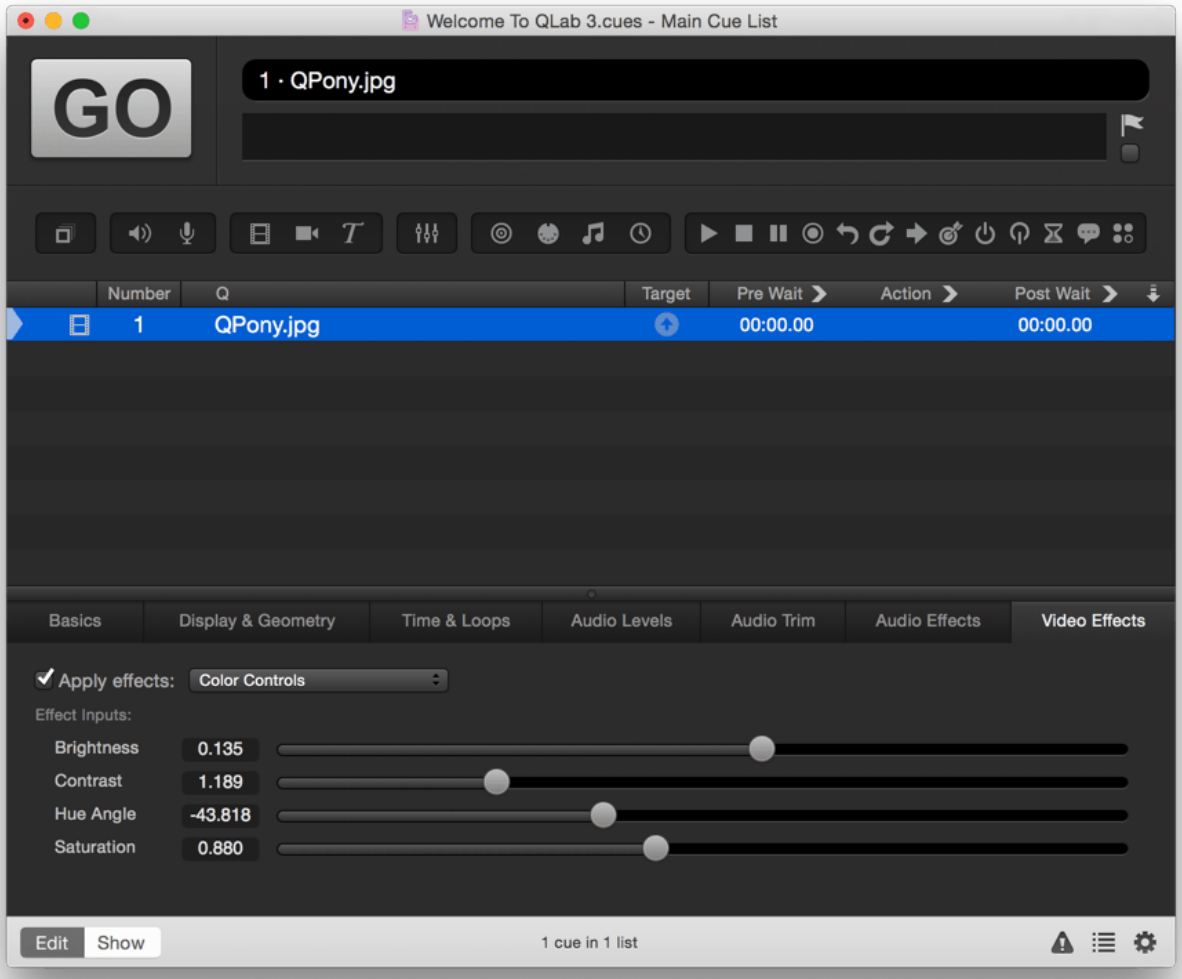
These three tabs allow you to control the audio associated with your Video cue. They operate the same as their corresponding tabs for Audio cues.

Video Effects

QLab gives you the ability to apply a wide range of live video effects to your Video cue. Please note that video effects can be extremely processor-intensive, and complicated effects have been known to bog down even very powerful Macs.

Only one video effect may be applied at a time.

Controls for each parameter of the effect will appear in the inspector; adjust each parameter individually using its appropriate control.



QLab comes with a variety of video effects pre-installed. You can also create your own effects using Quartz Composer. For a composition to be compatible

with QLab, it must be an effect (which is one of several types of compositions that Quartz Composer can create) and it must publish an input named `_protocolInput_Image` which is where QLab will send video to the composition, and `_protocolOutput_Image` which is where QLab will receive the effected video from the composition.

Broken Cues

Video cues can become broken for the following reasons:

Invalid video file.

Either the file is missing or damaged, or it's not one of the supported video file types.

No valid video surface assigned.

Assign a surface in the *Display & Geometry* tab of the inspector. You may also need to visit the Video section of Settings and create or adjust a surface to use with this cue.

There is a problem with this cue's video surface.

Visit the Video section of Settings and create or adjust a surface to use with this cue.

A license is required to send video to anything other than the default surface.

You'll need to either install a Basic Video, Pro Video, or Pro Bundle license, or reassign this cue to the default surface for this workspace.

A license is required to use custom geometry.

You'll need to either install a Pro Video or Pro Bundle license, or set this cue's mode to *Full Screen* in the *Display & Geometry* tab of the inspector.

A pro license is required to send video to a multi-screen surface.

You'll need to either install a Pro Video or Pro Bundle license, or reassign this cue to a surface which doesn't use multiple screens.

A pro license is required to send video to a partial screen.

You'll need to either install a Pro Video or Pro Bundle license, or reassign this cue to a surface which doesn't use partial screens.

A pro license is required to send video to a Blackmagic output device.

You'll need to either install a Pro Video or Pro Bundle license, or reassign this cue to a surface which doesn't use a Blackmagic output device.

A pro license is required to send video to Syphon output.

You'll need to either install a Pro Video or Pro Bundle license, or reassign this cue to a surface which doesn't use Syphon output.

A pro license is required to use video effects.

You'll need to either install a Pro Video or Pro Bundle license, or remove the video effect from this cue.

Timecode triggers require a Pro license.

You'll need to either install a Pro Audio or Pro Bundle license, or remove the timecode trigger from this cue.

Camera Cues

Camera cues, which require a Pro Video or Pro Bundle license, bring live video into QLab. Camera cues will recognize input from:

- Any IIDC-compliant webcam (note that it can often be frustratingly difficult to determine if a given webcam is IIDC-compliant.)
- Any Blackmagic DeckLink device (including DeckLink, UltraStudio, and Intensity devices)
- Syphon inputs (learn more at http://syphon.v002.info)
- Any FireWire DV camera

When a Camera cue is selected, three tabs will appear in the Inspector:

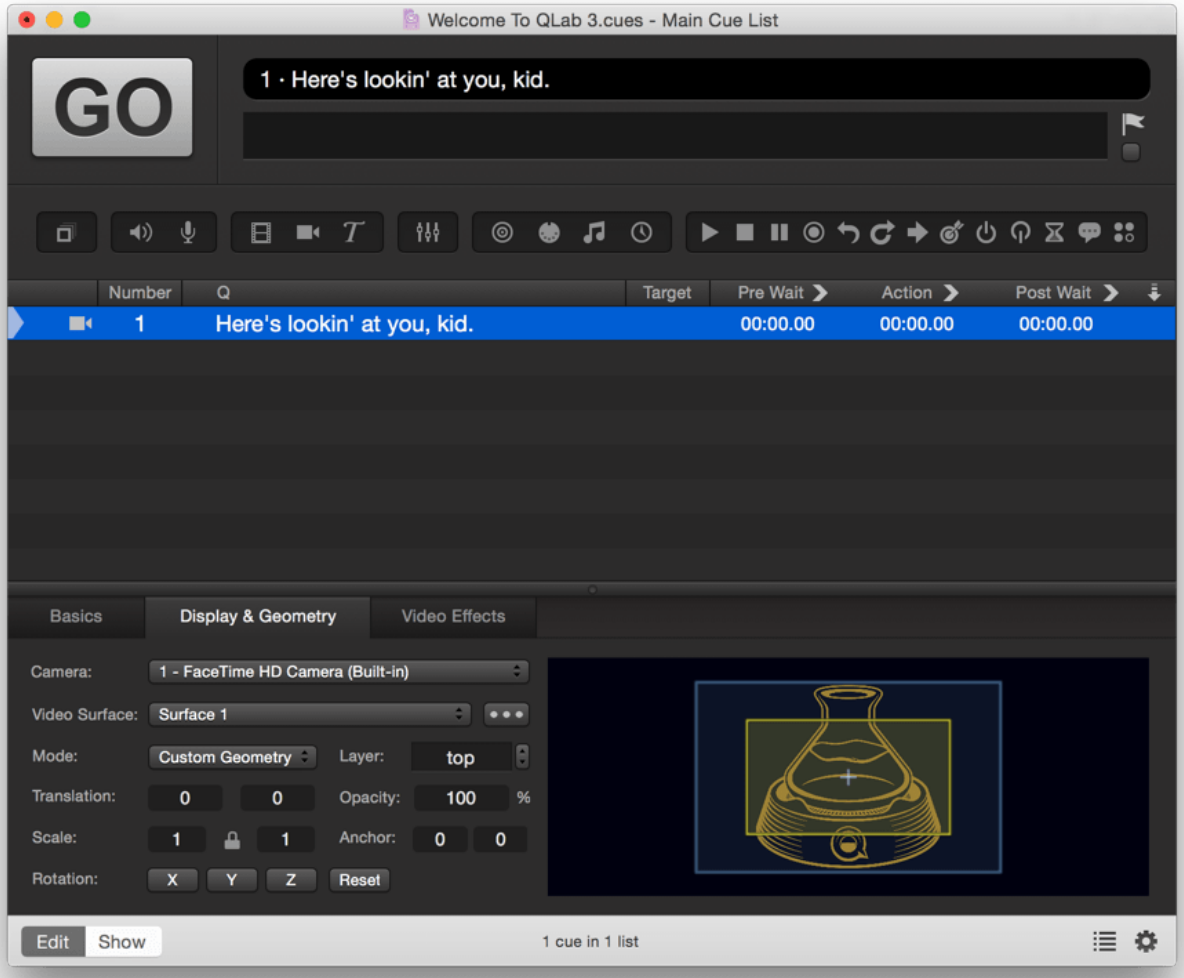
- Basics
- Display & Geometry
- Video Effects

Basics

Please refer to the section on the inspector in the **Getting Started** section of this documentation.

Display & Geometry

The *Camera* drop-down menu lets you choose from among eight camera patches, which are configured in the Camera section of Settings.



Once the camera input is chosen, Camera cues operate exactly the same way as Video cues. Note that Camera cues do not support audio.

Video Effects

Video effects for Camera cues operate the same as Video Effects for Video Cues.

Broken Cues

Camera cues can become broken for the following reasons:

No valid video surface assigned.

Assign a surface in the *Display & Geometry* tab of the inspector. You may also need to visit the Video section of Settings and create or adjust a surface to use with this cue.

There is a problem with this cue's video surface.

Visit the Video section of Settings and create or adjust a surface to use with this cue.

A license is required to send video to anything other than the default surface.

You'll need to either install a Basic Video, Pro Video, or Pro Bundle license, or reassign this cue to the default surface for this workspace.

A license is required to use custom geometry.

You'll need to either install a Pro Video or Pro Bundle license, or set this cue's mode to *Full Screen* in the *Display & Geometry* tab of the inspector.

A pro license is required to send video to a multi-screen surface.

You'll need to either install a Pro Video or Pro Bundle license, or reassign this cue to a surface which doesn't use multiple screens.

A pro license is required to send video to a partial screen.

You'll need to either install a Pro Video or Pro Bundle license, or reassign this cue to a surface which doesn't use partial screens.

A pro license is required to send video to a Blackmagic output device.

You'll need to either install a Pro Video or Pro Bundle license, or reassign this cue to a surface which doesn't use a Blackmagic output device.

A pro license is required to send video to Syphon output.

You'll need to either install a Pro Video or Pro Bundle license, or reassign this cue to a surface which doesn't use Syphon output.

A pro license is required to use video effects.

You'll need to either install a Pro Video or Pro Bundle license, or remove the video effect from this cue.

No valid camera source is assigned.

Assign a valid camera source in the *Display & Geometry* tab of the inspector. You may also need to visit the Camera section of Settings and connect a video input device to the desired patch.

A pro license is required to reactivate this saved cue.

You'll need to install a Pro Video or Pro Bundle license to use this cue.

Timecode triggers require a pro license.

You'll need to either install a Pro license, or remove the timecode trigger from this cue.

Titles Cues

download video ↓

Titles cues, which require at least a Basic Video license, allow you to display styled text in QLab, using any of the fonts installed on your Mac.

When a Titles cue is selected, four tabs will appear in the Inspector:

- Basics
- Display & Geometry
- Text
- Video Effects

Basics

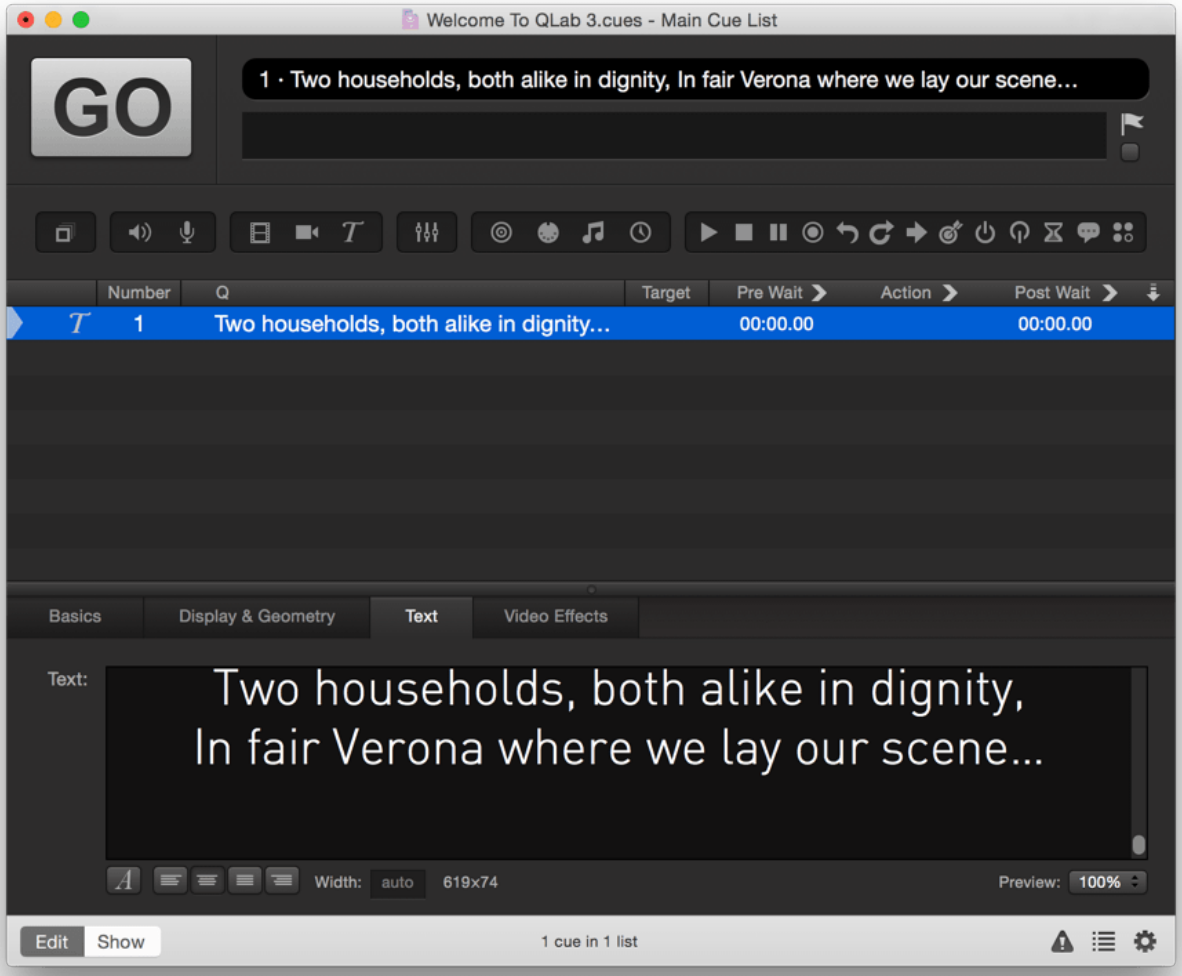
Please refer to the section on the inspector in the **Getting Started** section of this documentation.

Display & Geometry

Display & Geometry for Titles cues operates the same as Display & Geometry for Video Cues.

Text

The Text tab lets you input and style the text which will be displayed by the cue. The *A* button displays the Fonts panel which lets you adjust the font, size, color, and style of the selected text.



Next to that are four buttons which let you choose amongst left-justified, centered, full-justfied, and right-justified text.

When a Titles cue is playing, QLab renders the text as a PNG image and displays that image. Be default, the image dimensions are automatically calculated to fit the text exactly. The *Width* field allows you to manually increase the width of the (invisible) background, for example to allow necessary space for video effects to properly display. You can manually increase the height of the image by adding carriage returns above and below your text.

The *Preview* drop-down menu scales the display size of the text in the inspector to make it easier to edit at very small or very large font sizes.

Video Effects

Video effects for Titles cues operate the same as Video Effects for Video Cues.

Broken Cues

Titles cues can become broken for the following reasons:

No valid video surface assigned.

Assign a surface in the *Display & Geometry* tab of the inspector. You may also need to visit the Video section of Settings and create or adjust a surface to use with this cue.

There is a problem with this cue's video surface.

Visit the Video section of Settings and create or adjust a surface to use with this cue.

A license is required to send video to anything other than the default surface.

You'll need to either install a Basic Video, Pro Video, or Pro Bundle license, or reassign this cue to the default surface for this workspace.

A license is required to use custom geometry.

You'll need to either install a Pro Video or Pro Bundle license, or set this cue's mode to *Full Screen* in the *Display & Geometry* tab of the inspector.

A pro license is required to send video to a multi-screen surface.

You'll need to either install a Pro Video or Pro Bundle license, or reassign this cue to a surface which doesn't use multiple screens.

A pro license is required to send video to a partial screen.

You'll need to either install a Pro Video or Pro Bundle license, or reassign this cue to a surface which doesn't use partial screens.

A pro license is required to send video to a Blackmagic output device.

You'll need to either install a Pro Video or Pro Bundle license, or reassign this cue to a surface which doesn't use a Blackmagic output device.

A pro license is required to send video to Syphon output.

You'll need to either install a Pro Video or Pro Bundle license, or reassign this cue to a surface which doesn't use Syphon output.

A pro license is required to use video effects.

You'll need to either install a Pro Video or Pro Bundle license, or remove the video effect from this cue.

A video license is required to reactivate this saved cue.

You'll need to install a Basic Video, Pro Video, or Pro Bundle license to use this cue.

Timecode triggers require a Pro license.

You'll need to either install a Pro Audio or Pro Bundle license, or remove the timecode trigger from this cue.

Fading Video

A Fade cue can be used to adjust the opacity, translation, scale, rotation, video effect parameters, volume levels, and audio effect parameters of a targeted Video, Camera, or Titles cue. Fade cues can also target Audio cues and Mic cues; when a Fade cue is selected, the inspector will only show the tabs relevant to the type of cue that the Fade cue is targeting.

Fade cues require a target and a duration, and must adjust at least one level or parameter.

To learn how to set a target for a Fade cue, please refer to the section on targeting other cues in the **Getting Started** section of this documentation.

When a Fade cue which targets a Video, Camera, or Titles cue is selected, six tabs appear in the Inspector:

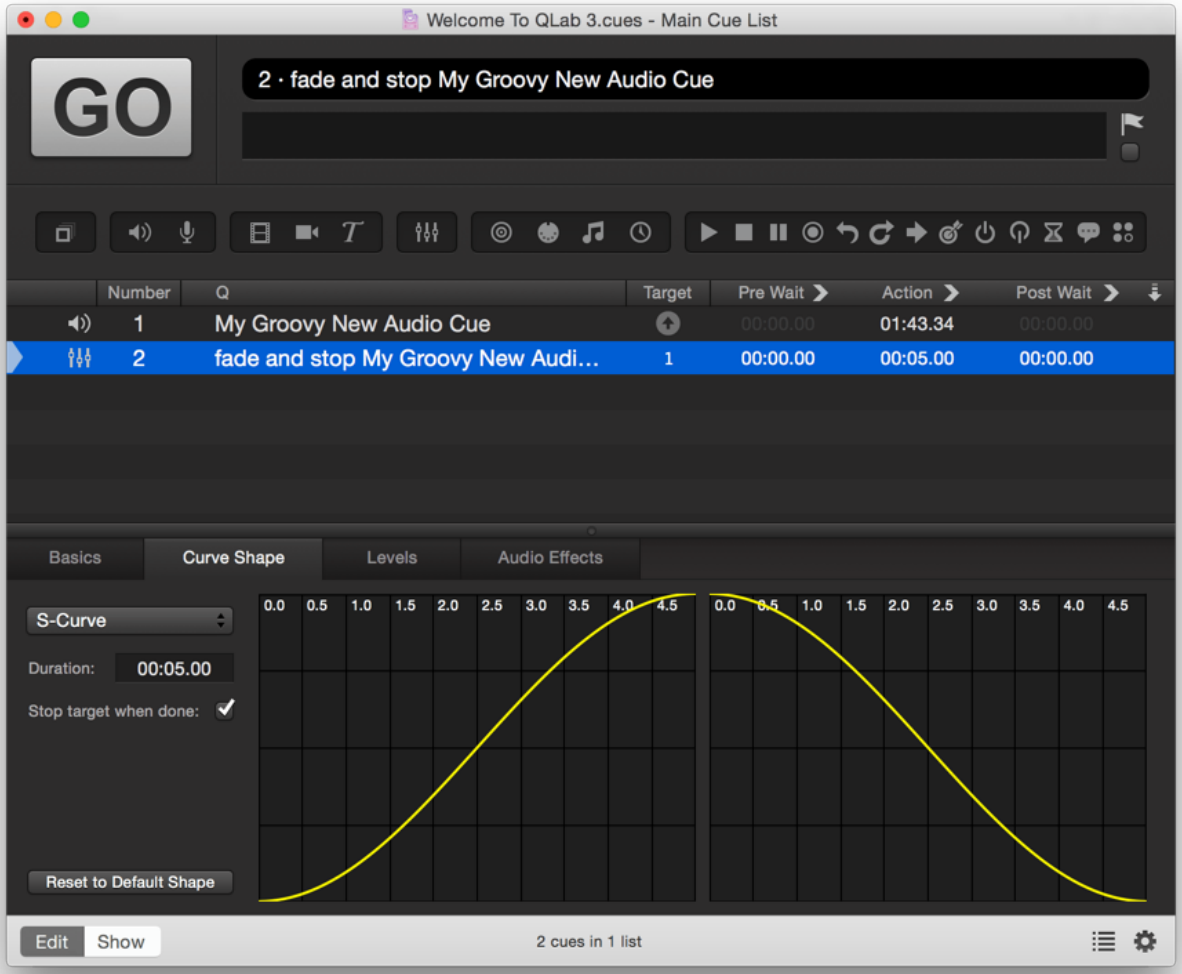
- Basics
- Curve Shape
- Levels
- Audio Effects
- Geometry
- Video Effects

Basics

Please refer to the section on the inspector in the **Getting Started** section of this documentation.

Curve Shape

The curve shape determines the manner in which the parameter or parameters are adjusted over the course of the fade. QLab defaults to an S-curve, but any curve can be drawn in the Curve Shape tab by selecting *Custom Curve* from the drop-down menu in the top left corner of the tab. The curve is drawn in yellow on the right side of the tab. The curve on the left is the shape for levels which are increasing, and the curve on the right is for levels which are decreasing. Click anywhere along the yellow line and a yellow dot, called a control point, will appear. Drag it to change the shape of the curve. Clicking on the curve again to create more control points. The active point will be filled-in yellow circle, and others are yellow outlines. To delete one, click on it to select it and press the `delete` key on your keyboard. To start over entirely, click *Reset to Default Shape* in the bottom left corner of the tab.



Check or uncheck the *Stop target when done* box under the Duration text field depending on whether you would like the target cue to continue playing after the Fade cue is complete, or stop once the Fade is complete.

Levels

The Levels tab allows you to specify which audio levels you wish to fade, and what their final volume will be. You can specify a different volume for each level.

For Camera cues and Titles cues, this tab has no effect.

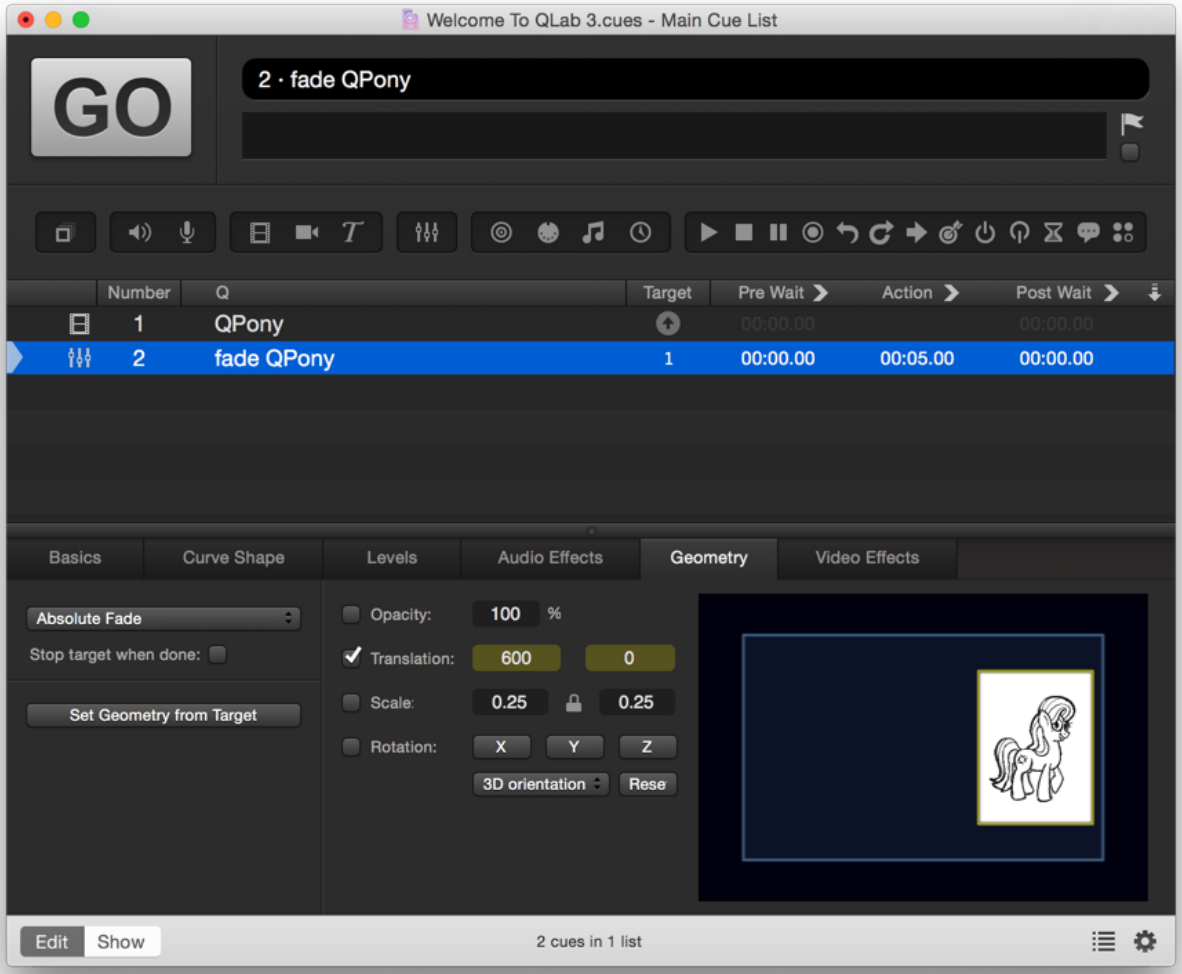
To learn more about the Levels tab in Fade cues, see the Fading Audio section of this documentation.

Audio Effects

See the Fading Audio Effects section of this documentation to learn about fading Audio Effecs.

Geometry

The Geometry tab allows you to specify which parameters of the Video cue you wish to fade, and what their final value will be.



By default, the checkbox next to each parameter will be unchecked, meaning that the Fade cue will not adjust that parameter. To fade a parameter with the Fade cue, just check the box next to that parameter.

Also by default, Fade cues are *Absolute*. This means that any parameters that you adjust with a Fade cue will arrive at their final levels regardless of their status before the Fade cue runs. If you use a Fade cue to set the scale of a target Video cue to 2, then the scale that Video cue will end up at 2 after running the Fade no matter what the scale was when the Video cue started off. If instead you set the Fade to be *relative,* then a Fade with a scale of 2 will double the current scale of the target Video cue.

Set Geometry from Target. This button will copy all geometry parameters from the target cue. You do not have to use this feature, but it can be convenient to give you a starting point for the Fade cue.

Opacity. You can fade the opacity of the target cue to any whole-number opacity between 0% and 100%.

Translation. You can move the target cue left to right and up to down by fading translation. Change the position of the video along the X-axis and/or Y-axis by entering values in the text fields or by clicking and dragging the thumbnail image to the desired spot in the preview box.

Scale. You can scale the target cue in both the X and Y axes together, or individually by clicking the padlock icon.

Rotation. Click and drag on each of the *X, Y,* and *Z* buttons to Fade the rotation of the target cue.

3D orientation. By default, Fade cues use three-dimensional quaternion to rotate their target. This results in smooth, natural movements when rotating along multiple axes. If you instead want to rotate the target a specific number of degrees about a single axis, you can use this drop-down menu to choose an axis, and then enter the number of degrees you wish to rotate.

Reset. Clicking the *Reset* button will zero out the rotation values in the Fade cue on all three axes, which means the Fade cue will rotate its target Video cue to the default “front facing” rotation.

Video Effects

When you apply an effect to an Video cue, any Fade cues that target that Video cue will recognize the effect you’ve applied. QLab will list all of the effect’s

parameters automatically under the Video Effects tab in the Fade cue.

By default, the checkbox next to each effect parameter will be unchecked, meaning that the Fade cue will not adjust that parameter. To fade a parameter with the Fade cue, just check the box next to that parameter.

Broken Cues

Fade cues can become broken for the following reasons:

No target cue.

Assign a target to the cue.

No fade parameters have been enabled; pick at least one thing to fade.

You can enable or disable an audio control by clicking in it; active controls will be highlighted in yellow. You can enable or disable a video control by checking or unchecking the box next to it.

One or more audio effects in this cue are missing or broken.

This typically happens when a workspace that uses audio effects is moved from one computer to another, and the new computer does not have the necessary AudioUnit installed. Either choose a different audio effect, or install the necessary AudioUnit.

A basic or pro video license is required to fade video geometry.

You'll need to either install a Basic Video, Pro Video, or Pro Bundle license, or adjust the Fade cue to not fade video geometry parameters.

A pro license is required to fade the playback rate.

You'll need to either install a Pro Audio, Pro Video, or Pro Bundle license, or adjust the Fade cue to not fade rate.

A pro audio license is required to fade audio effects.

You'll need to either install a Pro Audio, Pro Video, or Pro Bundle license, or adjust the Fade cue to not fade audio effects.

Timecode triggers require a Pro license.

You'll need to either install a Pro Audio or Pro Bundle license, or remove the timecode trigger from this cue.

Surfaces

QLab uses an abstraction called *Surfaces* to output video. A surface in QLab is a virtual video output which has one or more actual video output devices assigned to it. By creating QLab surfaces which represent the physical surfaces onto which you're projecting, QLab allows you to focus on the contents of your design rather than the mechanical details of your projection system.

Surfaces are designed to accommodate an extremely wide range of workflows and setups, from single screens, to video walls, to multi-projector blends on curved surfaces. As such, there's a lot of potential for complexity; the concept of surfaces allows that complexity to be isolated from the process of cue programming.

download video ↓

License requirements

The free version of QLab allows you to work with the simple case of outputting video to a single screen that you choose as a default surface for the workspace. The various advanced features introduced by QLab's surface setup require different licenses, as follows:

Feature	Free (no license installed)	Basic Video	Pro Video
Single-screen surfaces	One*	Unlimited	Unlimited
Multi-screen surfaces			✔
Screen layers	✔	✔	✔
Alignment guides	✔	✔	✔
Adjustment grids	✔	✔	✔
Edge blending			✔
Splits and warping			✔
Masks and constraints			✔
Screen rotation			✔
Partial screens			✔
Syphon output			✔

* *Multiple single-screen surfaces can be created without a license, but only the default surface can be used to play cues.*

Basic structure

As a rule, each surface in a QLab workspace corresponds conceptually to a physical surface on the stage. For example, you may be projecting onto two walls

and a door, all at different angles and all covered by one projector, in which case you will have three surfaces defined in your workspace (“Wall SL”, “Wall SR”, “Door”). Or you may have four projectors edge blended on a scrim, in which case you will have just one surface (“Scrim”). Each Video, Camera, or Titles cue you create is then assigned to a surface, and any geometry adjustments you make are relative to that physical surface.

Once a surface is defined, one or more screens can be assigned to it. Each surface may have more than one screen assigned, and each screen may be assigned to more than one surface.

QLab automatically detects when edge blending should happen and feathers the edges of each screen as appropriate, according to the following rules:

- If two screens overlap by more than 90%, they will be treated as double-stacked, and QLab will send the same image to both screens. For example, if you have two projectors that you want to combine to make a brighter image, overlapping them completely will tell QLab to mirror the output to the two projectors.
- If two screens overlap by more than 0% but less than 90%, QLab will automatically feather the overlapping edges, to create a seamless blend between projectors, unless you have disabled blending for the surface or its screens.
- If two screens do not overlap at all, the displays will render separate parts of the surface with no blending.

All your workspace’s surfaces are listed in the Video page of the workspace settings. Each has an *Edit* button, which opens the surface editor window, and an *X* button, which deletes the surface. Surfaces that are broken will appear with a yellow exclamation point, and cues routed to broken surfaces will also appear as broken.

A surface is defined as broken if it has no valid displays attached. If at least one assigned display is available, the surface will not report as broken, although it will alert you to any disconnected displays in the surface list. Surfaces may also appear as broken if they use features not supported by any of your currently installed licenses.

Auto-created surfaces

When you create a new workspace, QLab will automatically add a surface for each attached display, with that display assigned to the surface. This is for convenience, and intended to allow you to start programming cues quickly in simple setups. You are free to edit and delete these surfaces just as with any surface you create yourself. Any displays connected after a workspace is first created will need to be added manually to a new or existing surface.

Note: This is in contrast to QLab 3.0.x, which treated auto-created surfaces as special cases. Prior to 3.1, QLab would attempt to hide auto-created surfaces when displays were disconnected and create new ones when new displays were plugged in. This was problematic in cases where OS X was unable to differentiate consistently between identical displays.

Rendering pipeline

QLab's video rendering is a two-step process, with surfaces acting as the intermediate state between steps. First, all the cues playing to a surface render their frames, with any cue geometry and video effects applied, into an image buffer maintained by the surface. Then, that image buffer is used as the source for the final rendering step, in which each screen renders a portion of the full image with edge blending and warping applied.

One important consequence of this two-step rendering process is that surface order takes precedence over cue order. As such, each surface has a layer, much as video cues themselves do. In the first step, each surface renders all the cues patched to it, ordered by cue layer; in the second step, each screen renders every surface to which it's assigned, ordered by surface layer. If two surfaces occupy an overlapping portion of a screen, a cue on the top surface will show above the cue on the bottom surface, even if the cue layers would seem to indicate otherwise.

Creating and deleting surfaces

To create a new surface, click the *+* popup button below the surface list in the Video page of workspace settings. There are three workflows for creating a new surface:

1. Select *New Empty Surface* to create a totally blank surface, to which you can add screens manually.
2. Select *New With Display* to create a surface matching the resolution of a single display or partial display, with that display added to the surface's assigned screens.
3. Select *New Multi-Screen Surface* to quickly create a surface with a number of regularly arranged screens, such as a multi-projector blended surface or an LCD video wall.

Once a surface is set up, you can make any number of copies of it using the *Duplicate* button.

To delete a surface, click on the *X* button in its row in the list of surfaces.

The Surface Editor

Clicking the *Edit* button next to a surface brings up the surface editor in a separate window.

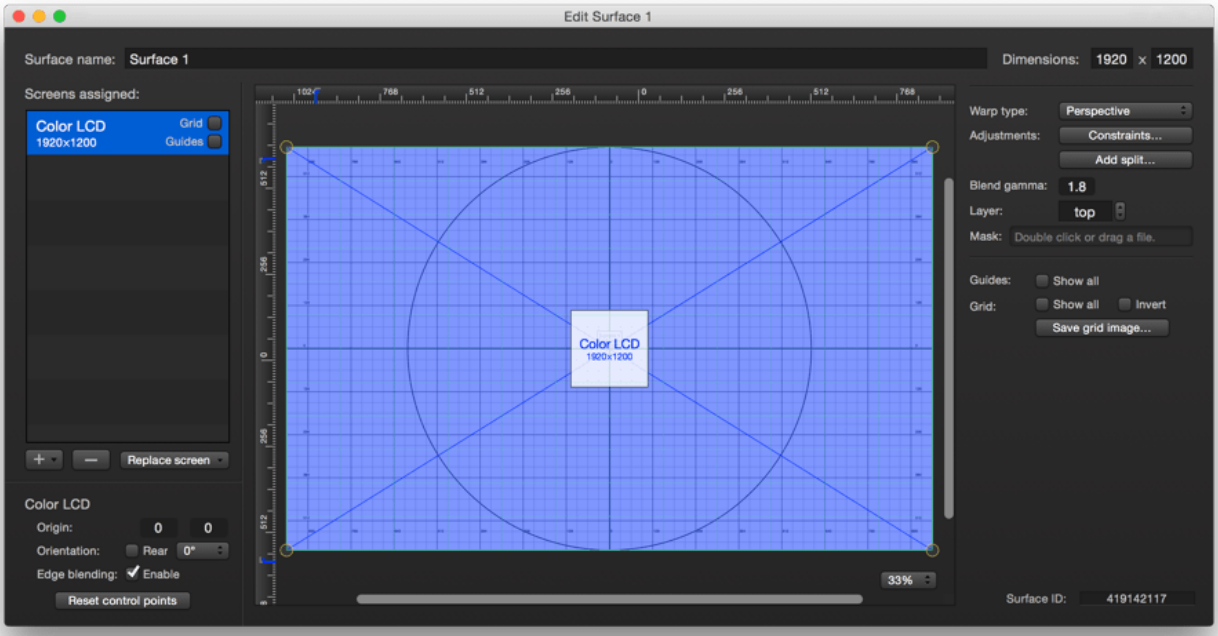
Many features visible in the Edit window require a Pro Video or Pro Bundle license, and will be disabled if no pro license is installed. The features allowed without a pro license will still function.

Below the header area (composed simply of the surface's name and dimensions in pixels), the editor window is divided into three columns:

The leftmost column shows a list of all displays assigned to the surface, as well as a section for information and controls pertaining to the currently selected display.

The center column is occupied by the surface canvas, where you can arrange screens on the surface and adjust control points for each screen.

The right column contains controls for parameters of the surface itself, as well as the currently selected control point or split (when applicable).



Left column: Displays

This area is where you add, remove, replace, and adjust the display(s) assigned to a surface. At the top is a list of all currently assigned screens, including the raster dimensions of each, and checkboxes to toggle the grid and guides for each.

To add a screen, click on the + button and select the screen to add. If you have a Pro Video or Pro Bundle license installed, you can also choose partial screens or Syphon. To remove the selected screen, click the - button.

Syphon output

Although Syphon appears in the list of screens, it does not show up visually in the canvas view of the surface. Syphon output simply mirrors the full surface (within any constraints), with no warping applied. You can add Syphon to any number of surfaces within a QLab workspace; each will create its own Syphon server, so each surfaces can be made available to other Syphon-enabled applications.

Adding Syphon output to a surface has no impact on processor load or performance, because each surface already uses a private Syphon server internally to send frames to the renderers that output video to each display. Adding Syphon to a surface simply means making that server public, so there is nothing additional happening within QLab. There is, of course, CPU and GPU load associated with whatever application is receiving the frames.

You can learn more about Syphon here.

Partial screens

Display splitters such as the Matrox DualHead2Go and TripleHead2Go and the Datapath x4 show up to OS X as a single, very large display. Using partial screens in QLab allows you to split that large display back up and address each output of the splitter as an independent virtual display. Partial screens can be arranged as 2-wide, 3-wide, 4-wide, or 2×2.

Partial screens require a Pro Video or Pro Bundle license.

Partial screens are designed exclusively for working with hardware splitters, and are not useful for making a surface cover only part of a projector. Instead, to restrict a surface to one part of a projector, use constraints, or simply reduce the surface dimensions and adjust the screen's origin.

Replace Screen

The *Replace screen* drop-down allows you to select a screen from the list, and then remap that screen to a different display. The drop-down lists only displays which are currently connected to your Mac. This is helpful if your hardware has changed, for example if you've prepared a show using an external monitor, then moved to the theater where you're using a projector. It can also help you if OS X mistakenly identified a screen as new, when in fact it's the same display you've been using. When you replace a screen, QLab updates the screen's raster

dimensions to match the new display, and any adjustments you've made such as control points are preserved as closely as possible.

Screen controls

The lower section of this column contains controls for the currently selected screen:

Origin controls determine (in X and Y pixels) the position of the screen's lower left corner relative to the lower left corner of the surface.

Orientation controls are composed of the *Rear* checkbox, which flips the screen's rendering horizontally for rear projection, and a *Rotation* pop-up, which rotates the image in 90° increments.

Edge blending can be disabled on a single projector (for example, if the projector has its own built-in blend controls) by unchecking this checkbox. This control is checked by default, but has no effect if the surface doesn't contain at least two partially overlapping screens.

The *Reset control points* button returns all the perspective/warping controls to their default locations. This applies to the currently selected screen only.

If the selected screen corresponds to a Blackmagic DeckLink-compatible device, a pop-up button will appear for selecting the resolution and framerate of the device output.

Center column: Editor canvas

The canvas shows a visual overview of the surface layout. With a Pro Video or Pro Bundle license installed, you can drag screens to arrange them on the surface,

drag splits to adjust them, and drag control points to adjust perspective correction and warping.

The canvas is bordered on the top and left by rulers that show surface measurements in pixels. These measurements are centered relative to the middle of the full surface, before any constraints.

The drop-down menu in the lower right corner of the canvas allows you to scale the view in the Surface Editor. It has no effect on QLab's output.

Right column: Surface controls

The controls in this column represent parameters of the surface itself.

Warp type

Warping controls require a Pro Video or Pro Bundle license.

Each screen assigned to a surface comes with a set of control points that you can use to warp the final rendering in various ways. This allows you to project a flat image onto an object that may be oddly shaped or at an angle to the projector, and have it appear perfectly aligned to the audience.

Control points are associated with individual screens, so warping adjustments made to one projector will not affect the image displayed by another projector on the same surface.

Note that it is generally best to arrange all the displays on a surface, and split the surface as desired, before adjusting control points.

QLab supports three different types of warping:

- Perspective (the default mode) is useful for simple corner pinning onto a flat surface. Any adjustments made are perspective-correct, meaning that once the corner pins are correctly placed, an image in one part of the surface will appear exactly the same size if moved to a different part of the surface. As a result, splits in Perspective mode surfaces may appear to have discontinuities.
- Linear is more useful for mesh warping, in conjunction with splits. Linear warping guarantees a continuous image across splits, with the tradeoff that it is not perspective-correct, and may distort images if used with extreme corrections.
- Bézier is a more complex mode, used to project onto curved surfaces. This mode gives you more control points, for full control of a 2D cubic Bézier surface. Like Linear mode, it is continuous across splits and useful for mesh warping.
- **Hint:** Bézier warps take longer to set up because of the additional control points; however, if you need to make a quick perspective tweak to a Bézier warp, you can do so by holding down the **Command** key while dragging one of the corner points. While **Command** is held down, QLab will interpolate the other points to approximate a perspective-correct corner pinning adjustment. This is not entirely precise, but it can be useful for minor tweaks.

Perspective correction is the most common use of QLab's warp controls. In the simplest case, a single projector directed at a flat screen may need corner pinning in order to compensate for skewing or keystoning caused by the angle of incidence of its beam.

To make corrections such as these, first check the *Grid* checkbox for the display being corrected, to display an adjustment grid on the projector. Then, in the canvas of the surface editor, drag the yellow control points at the corners until the grid displayed by the projector appears aligned. Fine-tune these

adjustments, if desired, by holding down the **Shift** key while dragging control points, or by adjusting the selected control point numerically using the *Selected control point* fields in the right column of the surface editor. When finished, uncheck the *Grid* checkbox to dismiss the grid.

In many cases, multiple projectors are edge-blended onto a single screen. Since each beam may hit the screen at a different angle, each projector has its own set of control points. It may be helpful to work with one projector at a time, adjusting control points with only a single grid shown, then to fine-tune the control points with all grids up and ensure that they overlap continuously.

Whenever possible, QLab attempts to adjust control points automatically to match other changes you make to a surface, such as moving screens or changing the layout of splits. However, once control point adjustments have been made, it is generally not possible to interpolate correct new locations for the control points when the surface setup changes. Rather than wildly guessing in these cases (since manual adjustment will be necessary anyway,) QLab leaves the control points in place for you to adjust. This can cause the image to appear distorted until you fix the control points, because they no longer match closely with the area they represent.

You may notice that the first adjustment you make to a screen's control point causes that screen's outline in the canvas to appear thicker. This indicates that an adjustment has been made, and the control points are fixed in place relative to the screen until you adjust them yourself. If you click on the *Reset control points* button, the thinner outline will return, and the control points will begin automatically adjusting to match the changes you make.

Usually, saving control point adjustments for the end of the process is the easiest way to ensure that moving screens and splits doesn't distort the image, and that the control points all start from a reasonable starting place. If that order of

operations is not possible, you can still adjust control points manually, or use the *Reset control points* button to start fresh.

Constraints

Constraints are a means of taking an existing surface and bringing its sides in to reduce its effective dimensions, while preserving all the warps and splits previously defined on the surface. The result, for the purposes of cue geometry, is simply a smaller surface.

To constrain a surface, click on the *Constraints...* button, and use the number fields in the popover to bring in each edge.

Constraints are most useful in conjunction with duplicating surfaces. For example, if you have a paneled wall defined as one surface, and want to define the individual panels as separate surfaces, you can first set up the full wall, along with any control point adjustments, then duplicate that surface and pull the duplicate's constraints in to isolate a single panel. This saves time compared to redoing all the corner pin adjustments to match the full wall's geometry perfectly.

Splits

Adding splits divides a surface up into "patches", each of which has its own set of control points. This opens up the possibility of everything from a surface spanning a single corner, to complex architectural projection mapping.

To add a split, click on the *Add split...* button. In the popover that appears, choose whether to split the surface horizontally (resulting in a left half and a right half) or vertically (into a top half and a bottom half), and provide the location for the split. You can set the location by clicking in the number field and dragging up and down. A thin dotted line will appear in the surface canvas as a preview, and will appear thicker once you click on *Add split* to commit it.

To move an existing split, click and drag it in the surface canvas. You can also adjust its positioning numerically by clicking to select it in the canvas, then using the *Selected patch* field.

To delete a split, click to select it in the canvas, then click the *X* button next to the *Selected patch* field.

Once a surface is split into patches, each patch is labeled with an identifier such as **A1**, **A2**, **B1**, etc. These labels show up in two places: The surface itself is displayed with large gray labels, and each display on the surface is shown with smaller, colored labels for patch identifiers. This allows you to determine which part of the surface corresponds to which set of control points, even if the control points are dragged to a different part of the editor than the surface patch they correspond to.

It is important to note that each set of control points on a screen corresponds only to the part of the surface patch they intersect, and not necessarily to the corners of the whole patch. For example, if a 1920 by 1080 surface is split evenly, at `x = 0`, into patches **A1** and **B1**, and the first projector ends just 10 pixels to the right of that split, then the control points labeled **B1** for that projector will control only that 10-pixel-wide strip, and not the whole 960-pixel width of patch **B1**.

If you move a screen until it no longer intersects with a patch at all, the number of control points associated with that screen will change. This can have unexpected results if you have control point adjustments in place, as it may change which patch a set of control points corresponds to. Again, saving control point adjustments for the end of the process will avoid this confusion.

Blend gamma

This adjusts the curve of the edge blending calculated by the surface. Adjust this if you see a dark or bright band running down the “seam” between projectors.

Surface Layer

The layer control for surfaces behaves just like the layer control for Video cues, but it's a separate control. So if one surface is set to layer 1, and another is set to layer 2, then *all* cues on the second surface will render on top of all cues on the first surface.

Mask

Surface masks are a way to completely or partially block certain pixels of a surface from displaying. They may be a familiar concept from graphics editing software, and they work similarly in QLab. Masks have several uses, including:

- Altering the shape of otherwise rectangular surfaces.
- Feathering the edges of a surface for a vignetting effect.
- Defining a region where another surface shows through from underneath.

To add a mask to a surface, double-click on the *Mask* well and select a file, or drag a file from Finder onto the well.

A mask should be a greyscale image that matches the dimensions of the surface, but QLab will convert it to greyscale and re-size it to fix the surface if necessary.

Black pixels in the mask are regarded as opaque, white pixels are regarded as transparent, and grey pixels are partially transparent; the darker the grey, the less transparent they are.

Any standard image format can be used as a surface mask. There is no performance difference between formats when rendering because QLab re-renders the image when importing it for use as a mask. However, if the size of the mask image is a different from the size of the surface, QLab must scale it "live" and that can negatively affect performance, so try to match your surface dimensions whenever possible.

QLab does not have an integrated mask editor. Instead, it watches the mask image file to look for changes; when it sees that the file has changed, it automatically reloads it. Thus, you can use any graphics editing program to make adjustments to a mask, and see changes applied live as you go.

Guides

The *Guides* checkbox toggles a set of lines and targets for projector alignment. With the basic layout of screens on a surface in place, you can use the guides to "rough in" the projectors to cover the desired physical area on stage.

Red guides show you the edges and center of each projector, and a green dotted line shows where the overlap with each adjacent projector should end. Guides show the bounds of the full projector raster, and are not affected by control points.

Grid

Use the *Grid* checkbox to send a black-and-white grid image to each projector while you adjust control points. The grid's dimensions match those of the surface, and only the portion corresponding to the area covered by each projector is output. In other words, it is essentially equivalent to running a video cue with a still image of a grid in full-screen mode, except that it can be sent to individual screens in isolation. When all corner points are in place and grids for all screens are enabled, you should see a single, perspective-correct, continuous grid for the whole surface.

By default, the grid is rendered black-on-white; use the *Invert* checkbox to switch it to white-on-black.

The *Save grid image...* button creates an image file that can serve as a useful reference when constructing or fine-tuning masks in a graphics program.

Below the *Grid* controls are fields for numerically adjusting the selected control point or split. When none is selected, these controls are hidden.

Surface ID

When a surface is created, QLab gives it a unique surface ID number. This number can be useful when using OSC or AppleScript to interact with surfaces. The surface ID is displayed in the lower right corner of the Surface Editor for easy reference and copying.

Chapter 5: Control

OSC Cues

OSC cues allow QLab to send messages using the Open Sound Control protocol; a flexible, extensible, network-based messaging system designed as a sort of successor to MIDI.

When an OSC cue is selected, two tabs will appear in the Inspector:

- Basics
- Settings

Basics

Please refer to the section on the inspector in the **Getting Started** section of this documentation.

Settings

Destination. This drop-down menu shows the sixteen OSC patches, or destinations, to which QLab can send OSC messages. You can edit this list in OSC Settings.

Message Type. Select which type of message you want to send.

- **QLab OSC Message.** For convenience when using OSC cues to communicate with another computer running QLab, this option provides a very simple interface for sending several commonly used commands. Enter a cue number below, and choose a command to send to the remote copy of QLab.
- **Custom OSC Message.** This option gives you a text field to enter any OSC command. For example, to tell a remote copy of QLab to GO, enter the command `/go`. Multiple arguments in a custom OSC message are separated by spaces, like this:

```
/my/groovy/message 2 10 12
```

That sends message `message` to address `/my/groovy` with three integers, `2`, `10`, and `12`, as separate arguments.

- **Raw UDP String.** This option allows you to send strings over UDP which will not be encoded as OSC by QLab on the way out. Some software, such as Medialon Manager, can accept commands as plaintext strings over UDP, and while this isn't technically OSC, we use the OSC cue as the place to achieve this.

Send Message. Click this button to test-send your message.

Broken Cues

OSC cues can become broken for the following reasons:

Invalid network destination.

Assign a valid network destination in the *Settings* tab of the inspector. You may also need to visit the OSC section of Settings and specify a valid destination

address and port for the desired patch.

Missing QLab cue number.

Fill in a valid cue number in the *Settings* tab of the inspector. This is only relevant for OSC cues set to the *QLab OSC message* type.

Invalid command parameters.

Fill in valid parameters in the *Settings* tab of the inspector. This is only relevant for OSC cues set to the *QLab OSC message* type.

No OSC message specified.

Fill in a valid OSC message in the *Settings* tab of the inspector. This is only relevant for OSC cues set to the *Custom OSC message* type.

Invalid OSC message.

Fill in a valid OSC message in the *Settings* tab of the inspector. This is only relevant for OSC cues set to the *Custom OSC message* type.

No UDP message specified.

Fill in a valid UDP string in the *Settings* tab of the inspector. This is only relevant for OSC cues set to the *Raw UDP string* type.

A license is required to reactivate this saved cue.

You'll need to install a license in order to use this cue.

MIDI Cues

MIDI cues allow you to send MIDI voice messages, MIDI Show Control (MSC) messages, or MIDI System Exclusive (SysEx) messages.

When a MIDI cue is selected, two tabs will appear in the Inspector:

- Basics
- Settings

Basics

Please refer to the section on the inspector in the **Getting Started** section of this documentation.

Settings

MIDI Destination. This drop-down menu shows the eight MIDI patches, or destinations, to which QLab can send MIDI messages. You can edit this list in MIDI Settings.

Message Type. Choose among MIDI Voice Message, MIDI Show Control Message, and MIDI SysEx Message.

MIDI Voice Message ("Musical MIDI")

Command. QLab can send the following types of MIDI Voice commands: Note On, Note Off, Program Change, Control Change, Key Pressure (Aftertouch), Channel Pressure, and Pitch Bend Change. All commands require a channel, but the other controls available will vary depending on the type of command selected.

Control Change, Key Pressure, Channel Pressure, and Pitch Bend commands can be faded from one value to another over time. If you select any of these options, a set of controls will appear to enable fading.

To fade a message, check the box marked *Fade over duration* and enter the desired duration in the field provided.

You can also adjust the curve of the fade using the graph on the right.

Because MIDI cues have no way of knowing the current value of the parameters that they are controlling, you need to enter both a starting value, which is the field labeled *Value* or *Velocity* underneath the Command drop-down menu, and an ending value, which is the field labeled *Fade to value* or *Fade to velocity* towards the middle of the inspector.

MIDI Show Control Message (MSC)

Command Format. The MSC spec defines a number (quite a large number) of specific categories within which devices or software can self-identify. Choose the category for the receiving device here.

Command. Select the MSC command you wish to send here.

Device ID. Enter the MSC device ID number of the receiving device here. Device ID 127 is the “all-call” ID, and all MSC devices on your MSC network will respond to the message.

Q Number, Q List, and **Q Path** should be filled in according to the needs of the receiving device.

MIDI SysEx Message

Enter your SysEx message here in hexadecimal format. Omit the leading `F0` and `F7`; QLab adds those for you.

Send Message. Click this button to test-send your message.

Broken Cues

MIDI cues can become broken for the following reasons:

No MIDI destination.

Choose a MIDI destination in the *Settings* tab of the inspector. You may also need to visit the MIDI section of Settings and connect a MIDI device to the desired patch.

Illegal characters used in SysEx message.

Edit the SysEx message in the *Settings* tab of the inspector.

Length of SysEx message is invalid.

Edit the SysEx message in the *Settings* tab of the inspector.

A license is required to reactivate this saved cue.

You'll need to install a license in order to use this cue.

MIDI File Cues

MIDI File cues contain a sequence of MIDI events and play them to a single MIDI output. While QLab does not have a native concept of tempo and meter maps, MIDI File cues provide a way to create individual timelines on which events occur with tempo- and meter-based timings. This can be useful when synchronizing MIDI-triggered events with a piece of music, for example.

Files supported

The MIDI File cue supports both Type 0 (single-track) and Type 1 (multitrack) standard MIDI files. Type 2 files (an uncommon multiple-sequence format) are not supported.

When a MIDI File cue is selected, two tabs will appear in the Inspector:

- Basics
- Settings

Basics

Please refer to the section on the inspector in the **Getting Started** section of this documentation.

Settings

MIDI Destination. This drop-down menu shows the eight MIDI patches, or destinations, to which QLab can send MIDI messages. You can edit this list in MIDI File Settings.

The MIDI File cue supports any number of tracks, and events can occur on any MIDI channel (1-16), but all messages in the file are sent to a single MIDI port. This can be a physical port connected to a tone generator, lighting board, or other gear, or a software MIDI destination such as ipMIDI. It can also be an IAC bus, which allows MIDI events to loop back into QLab and trigger other cues, or to route to another application running on the same computer.

Playback. This field is a multiplier for all tempi in the MIDI file. A rate of 0.5, for example, will result in half-speed playback.

MIDI File cue timings are based on the computer's internal clock, and are not clocked to any audio or video devices.

Broken Cues

MIDI File cues can become broken for the following reasons:

Invalid MIDI destination.

Choose a MIDI destination in the *Settings* tab of the inspector. You may also need to visit the MIDI File section of Settings and connect a MIDI device to the desired patch.

Invalid MIDI File.

Either the file is missing or damaged, or it's not a valid MIDI file.

A license is required to reactivate this saved cue.

You'll need to install a license in order to use this cue.

Timecode Cues

Timecode cues comprise QLab's mechanism for generating outgoing timecode.

Unlike most timecode-enabled applications which have a single, fixed timeline from which timecode can be generated directly, or which can be driven by incoming timecode, QLab allows for multiple independent timelines running concurrently. By encapsulating timecode output into a cue, QLab can generate multiple independent timecode streams to drive other devices.

When a Timecode cue is selected, two tabs will appear in the Inspector:

- Basics
- Settings

Basics

Please refer to the section on the inspector in the **Getting Started** section of this documentation.

Settings

Type. Timecode cues can generate either MIDI Timecode (MTC) or Linear Timecode (LTC). Depending on which type you select here, the inspector will show different output options.

LTC (Linear, or Longitudinal, Timecode) is meant to be carried on a line-level audio connection. When LTC is selected, the **Destination** controls allow you to select an audio device and a channel number to output to. The channel number defaults to 0, which is not valid, so a channel number valid for the selected device must be specified.

> **Warning:** Use caution when setting up the routing of an LTC signal, from QLab or any source. LTC contains strong high-frequency components which, if accidentally routed out to speakers, can cause damage to hearing and to friendships.

MTC (MIDI Timecode) carries most of the same information as LTC, but on a MIDI connection rather than audio. This can be a physical MIDI port, or an IAC bus. No channel number is required, as MTC messages are not associated with a MIDI channel.

> Sending MTC over a network is discouraged, as the inconsistent latency of a network connection can often render the timecode signal inaccurate, or even unreadable, on the receiving end. Inexpensive MIDI interfaces often create the same problem, even those that work well for normal musical MIDI. Always use a reliable, high-quality MIDI interface when working with MTC.

Framerate. QLab supports both **video speed** and **film speed** options at all common framerates. Be sure to match the framerate on the receiving end precisely.

Broadly speaking, “film speed” refers to the timecode formats with integer framerates (24, 25, 30 drop, 30 non-drop), while “video speed” refers to their non-integer counterparts (23.976, 24.975, 29.97 drop, and 29.97 non-drop).

Each video speed framerate is an identical timecode format to its film speed counterpart, but pulled down by 0.1%. For example, a frame of 29.97 non-drop timecode consists of the same data as the same frame of 30 non-drop, but at a 0.1% slower rate. Neither LTC nor MTC differentiates between video speed and film speed in how the bits are encoded, so timecode at the wrong speed will initially appear correct on the receiving end. However, the timecode will drift noticeably over time from what is expected unless the speeds match.

Timecode cues are clocked differently depending on the type selected. LTC follows the clock of the audio device to which it outputs, and is guaranteed not to drift from that clock. MTC, on the other hand, follows the computer's internal clock. Under normal use, drift between high-quality devices is usually minimal, but if drift-free synchronization with another machine is required over long stretches of time, the best option is to output LTC to an audio device that can resolve to the same master clock (via word clock, etc.) as the other machine.

Start time. This control allows you to specify the first frame of timecode that is transmitted when the cue is triggered. Bear in mind that both LTC and MTC can take up to a few frames to transmit enough information to read. If an event needs to be triggered on a specific frame, it is best to start timecode output a few frames ahead, as preroll into that event. This is why the default start time for a Timecode cue is `1:00:00:00` rather than `0:00:00:00`. Because there is no room for preroll before hour 0, the best practice is to treat hour 1 as the beginning of the timeline, with space for preroll beforehand.

Broken Cues

Timecode cues can become broken for the following reasons:

No MIDI destination.

Choose a MIDI destination in the *Settings* tab of the inspector. You may also need to visit the MIDI section of Settings and connect a MIDI device to the desired patch. This is only relevant to Timecode cues set to the *MTC* type.

Invalid LTC audio channel.

Choose an audio device and channel in the *Settings* tab of the inspector. You may also need to visit the Audio section of Settings and connect a MIDI device to the desired patch. This is only relevant to Timecode cues set to the *LTC* type.

A Pro license is required to reactivate this saved cue.

You'll need to install a Pro Audio, Pro Video, or Pro Bundle license in order to use this cue.

Script Cues

Script cues allow you to execute AppleScript from within QLab.

When a Script cue is selected, two tabs will appear in the Inspector:

- Basics
- Script

Basics

Please refer to the section on the inspector in the **Getting Started** section of this documentation.

Script

The text field in the Script tab will show the default AppleScript defined in Script Settings.

Text entered here will follow Apple's standard formatting rules for AppleScript.

Compile Script. Click here to compile your script. If your script shows errors, QLab will display an error message next to this button in order to help you find and solve the error.

Run in separate process. By default, QLab spins off the script in a Script cue to an external invisible application which handles the execution of the AppleScript.

This allows complex scripts to execute in the background, without monopolizing QLab and preventing you from interacting with QLab while the script runs. If your script, for some reason, needs to execute from within QLab, uncheck this box.

Broken Cues

Script cues can become broken for the following reasons:

Empty script.

Fill in a valid script in the *Script* tab of the inspector.

AppleScript error

Correct the error in the *Script* tab of the inspector.

A license is required to reactivate this saved cue.

You'll need to install a license in order to use this cue.

Other Cues

Start, Stop, and Pause

Start, Stop, and Pause cues have no inspector tabs other than the Basics tab. They have only a target, which must be another cue in the workspace. Each of these three cue types has a single function:

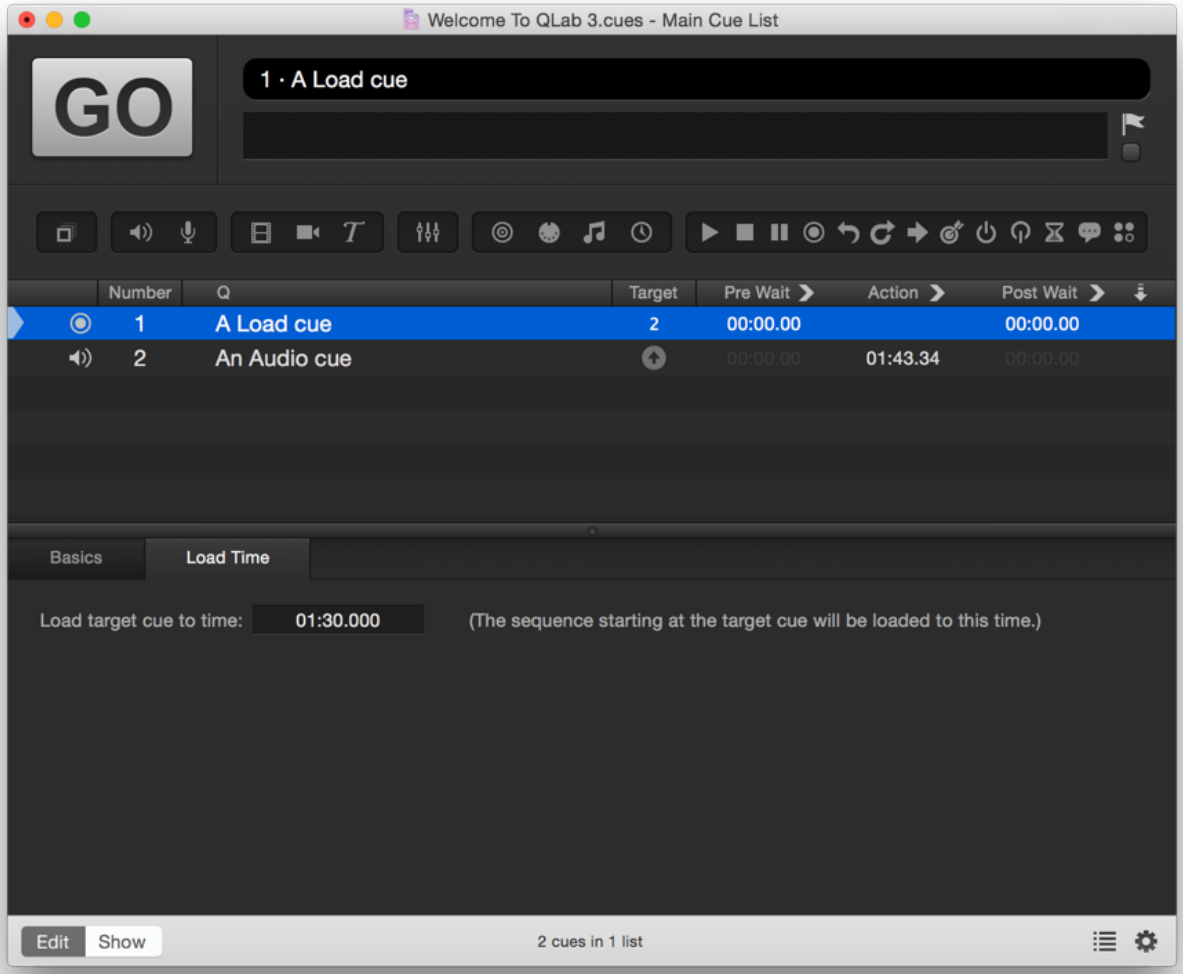
- A Start cue starts its target cue. Note that it does not move the playback position.
- A Stop cue stops its target cue.
- A Pause cue pauses its target cue. Use a Start cue to resume.

Broken Cues

Start, Stop, and Pause cues will only become broken if they have no valid target cue.

Load

The Load cue loads its target cue. If the target cue has a non-zero action time, then you can load that target cue to a specific time via the Load Time tab in the inspector.

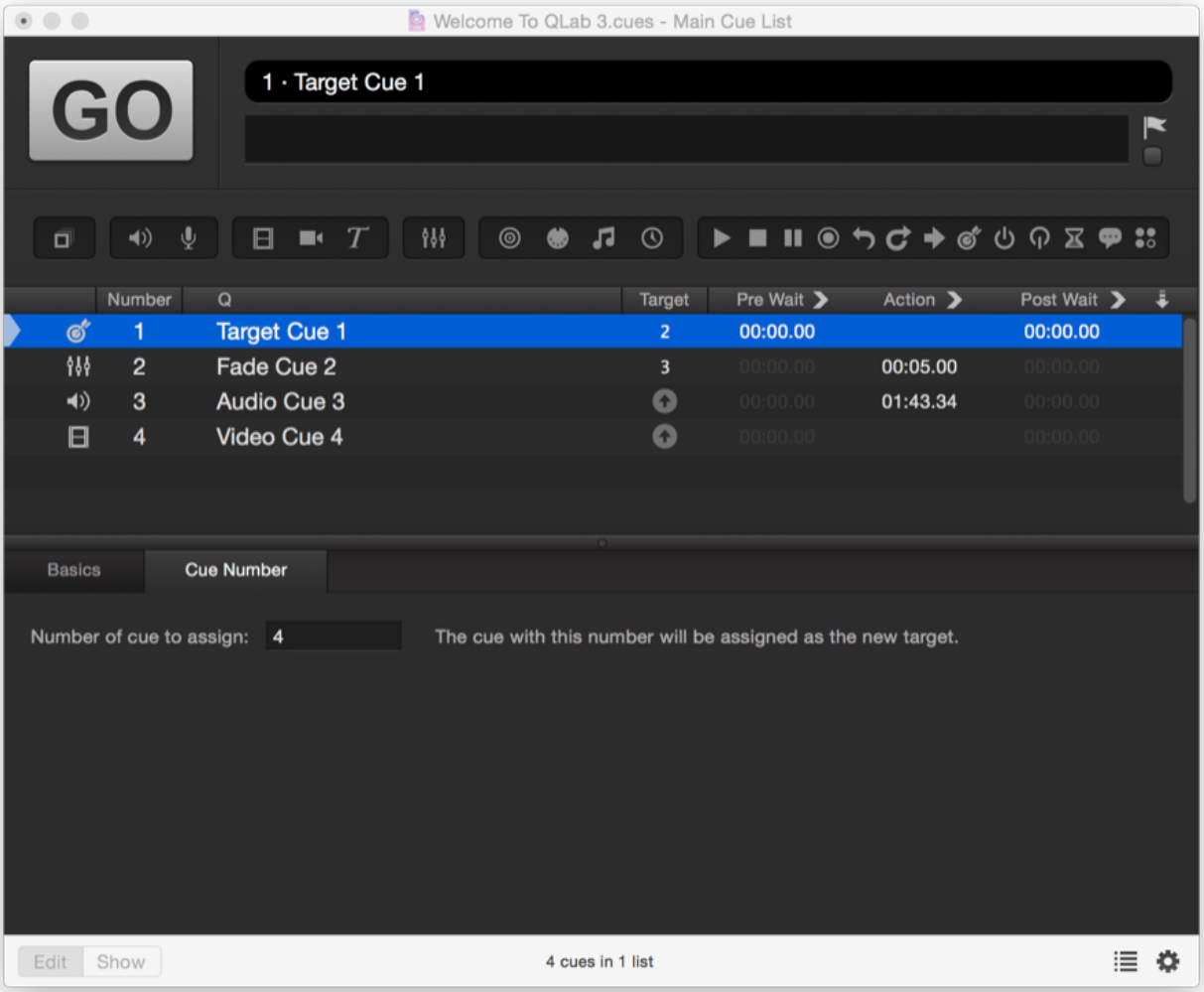


Broken Cues

Load cues will only become broken if they have no valid target cue.

Target

The Target cue changes the target of another cue in the workspace.



In the above screen shot, Target Cue 1 targets Fade Cue 2, and sets Fade Cue 2's target to Video Cue 4.

Broken Cues

Target cues can become broken for the following reasons:

Invalid target cue number.

There is no cue in this workspace with the given number.

Assign a valid cue number in the *Cue Number* tab of the inspector.

No target cue.

Assign a valid target cue.

Reset

The Reset cue has no inspector tabs other than the Basics tab. It requires a target, and when a Reset cue is triggered, it resets any temporary changes made to its target.

This begs the question: what are temporary changes?

The first and perhaps most obvious answer is that a change of target caused by a Target cue is a temporary change, and using a Reset cue will revert the change made by a Target cue. Additionally, there are a number of OSC methods which make changes to the live state of a cue, which is to say they only have an effect if the cue is currently running. You can use a Reset cue to revert these temporary changes.

Broken Cues

Reset cues will only become broken if they have no valid target cue.

Devamp

The Devamp cue is discussed as part of the Slices and Vamping section of this documentation.

Broken Cues

Devamp cues can become broken for the following reasons:

A license is required to reactivate this saved cue.

You'll need to install a license to use this cue.

No target cue.

Assign a valid target cue.

GoTo

The GoTo cue has no inspector tabs other than the Basics tab. It requires a target, and when a GoTo cue is triggered, QLab moves the playback position to that target cue. Note that the target cue is not automatically started; in this respect, the GoTo cue is a complementary cue to the Start cue.

Broken Cues

GoTo cues will only become broken if they have no valid target cue.

Arm and Disarm

Arm and Disarm cues have no inspector tabs other than the Basics tab. Their only role is to arm and disarm their target cues, respectively.

Broken Cues

Arm and Disarm cues will only become broken if they have no valid target cue.

Wait

The Wait cue has no inspector tabs other than the Basics tab, and no settings other than Action. A Wait cue's post-wait time is automatically set to be equal to its duration. You can use a Wait cue in combination with auto-follows or auto-continues as an alternate way to create cue sequences, or as a simple timer for tasks outside of QLab.

Memo

The Memo cue has no inspector tabs other than the Basics tab, and has no effect when triggered. You can use Memo cues as a place to store notes to your operator (as the name of the Notes cue, or in the cue's notes field), as a visual separator between other cues, or for some similar reason.

Chapter 6: Scripting

OSC Dictionary

QLab has an extensive API (application program interface) for OSC which allows you to control QLab from any device or software which can broadcast OSC messages. What follows here is a complete dictionary of QLab's OSC implementation.

Transport layer

The QLab OSC API can be used over both UDP and TCP transport layers. QLab listens for incoming OSC on port 53000.

When talking to QLab via UDP, each OSC message corresponds to one UDP datagram. Replies to OSC via UDP are sent on port 53001.

When talking to QLab via TCP, messages are framed using the double END SLIP protocol (RFC 1055) as required by the OSC 1.1 specification. Replies to OSC via TCP are sent on port 53000.

QLab also listens for plain text on UDP port 53535, and attempts to interpret it as OSC. For example, sending the text `/cue/selected/start` to QLab on UDP port 53535 will have the same result as sending the actual OSC command **/cue/selected/start** to port 53000.

The OSC API behaves almost identically when using both UDP and TCP. Exceptions are noted below, such as cases where a reply may be larger than the maximum size of a UDP datagram.

Reply format

All replies from QLab take the form:

```
/reply/{/invoked/osc/method} json_string
```

The reply is sent to the IP address from which the original message was received.

The address of the reply starts with `/reply/` and ends with the address of the invoked method.

`json_string` takes the form:

```
{
    "workspace_id" : string,
    "address": "/invoked/osc/method",
    "status": string,
    "data": value
}
```

`workspace_id` is optional, and only specified if the reply is specifically from the given workspace.

`status` is either `ok` or `error`.

`data` is the JSON-encoded result of invoking the method at `address`

For example, a workspace `cueLists` method:

```
/workspace/34200B51-835A-4918-A137-B6511784B6CA/cueLists
```

would cause QLab to respond with:

```
/reply/workspace/34200B51-835A-4918-A137-B6511784B6CA/cueLists {json_string}
```

Update format

When a client has requested updates from a specific workspace (i.e. has sent that workspace the `/workspace/{id}/updates 1` command), it will receive push notifications when the client needs to update state for a cue or workspace. The client may receive the following messages at any time:

```
/update/workspace/{workspace_id}
```

This message is sent if you need to reload the cue lists for the workspace.

```
/update/workspace/{workspace_id}/cue_id/{cue_id}
```

This message is sent if you need to reload the state for the given cue. If the cue is a group cue or cue list, you should also reload the children of the cue.

```
/update/workspace/{workspace_id}/cueList/{cue_list_id}/playbackPosition {cue_id}
```

This message is sent when the playback position of {cue_list_id} changes to {cue_id}. If there is no current playback position, there will be no {cue_id} argument.

```
/update/workspace/{workspace_id}/disconnect
```

This message is sent if you need to disconnect from the given workspace (because it is going away).

Application methods

QLab will respond to the following general commands:

/version

Returns QLab's version number.

/alwaysReply {boolean}

By default, QLab will only send a reply if the method generates a reply to send.

However, if alwaysReply is set to any non-zero number, QLab will send a reply for every OSC message it receives. Messages that would not normally generate a reply will generate one with a JSON string argument that contains:

```
{
    "workspace_id" : string,
    "address": "/invoked/osc/method",
    "status": string
}
```

The `status` string will be either "ok" or "error".

Note that the `data` field does not exist in this reply.

/workspaces

Returns an array of workspace dictionaries:

```
[
    {
        "uniqueID": string,
        "displayName": string
        "hasPasscode": number
    }
]
```

/toggleAuditionWindow

Show or hide the audition window.

/auditionWindow {number}

Number is interpreted as a boolean, and sets whether the audition window is visible. If no argument is provided, this returns the current visibility of the audition window.

/workingDirectory {path}

Get or set the current working directory. If provided, the `path` string is the working directory you wish to set. You can provide two kinds of paths:

- Full paths, e.g. `/a/full/path/to/some/directory/`
- Paths beginning with a tilde, e.g. `~/a/path/to some/directory`

Paths beginning with a tilde (~) will be expanded; the tilde signifies “relative to the user’s home directory”.

Current workspace methods

The current workspace is the front-most, active document. All Workspace Methods can be sent to the current workspace without specifying its uniqueID. To do this, specify the workspace method without the `/workspace/{id}` portion of the address.

For example, the following commands would be applied to the current workspace:

```
/connect {passcode_string}

/disconnect

/go

/stop

/undo
```

etc.

The full set of Workspace Methods is described below.

Workspace methods

Workspaces may be addressed either by their display name or their unique ID. For all OSC methods shown below of the form `/workspace/{id}/...` the `id` may be either the display name of the workspace, such as `MyGroovyWorkspace.cues`, or the unique ID of the workspace, which can be found in the Info tab of the Status Window.

Note, however, that addressing by display name will work only if the display name is composed of characters allowed in OSC method names. This does NOT include spaces, unicode, diacritical, or other “special” characters.

Addressing a workspace by its display name looks like this:

```
/workspace/demo.cues/cue/1/colorName green
```

Individual workspaces (documents) will respond to the following commands:

/workspace/{id}/connect {passcode_string}

Connect to this workspace with an optional passcode string. If the workspace has a passcode, you MUST supply it before any other commands will be accepted by the workspace or the cues it contains. If the workspace does not have a passcode, the `/workspace/{id}/connect` method is optional.

Returns `ok` if there is no passcode, or the passcode matches.

Returns `badpass` if the passcode does not match.

Returns `error` if the workspace does not exist.

/workspace/{id}/disconnect

Disconnect from this workspace. You should invoke this method when you will no longer be sending messages to this workspace.

If you are communicating to QLab via UDP, QLab will automatically disconnect your client if it has not heard any messages from it in the last 31 seconds. Any message (e.g. “thump”) will serve to keep the client connected. If you are disconnected and the workspace has a password, you will need to reconnect with that password before further commands will be accepted.

If you are communicating to QLab via TCP, QLab will not automatically disconnect your client. The client will remain connected until the client sends the disconnect message or the TCP connection itself is disconnected.

/workspace/{id}/thump

A simple heartbeat method for this workspace. Returns the data “thump”. (Thump-thump. Thump-thump.)

/workspace/{id}/updates {number}

`number` is interpreted as a boolean. If yes, your client wants push notifications of cue changes. If no, your client no longer wants push notifications of cue changes.

/workspace/{id}/save

Tell the given workspace to save itself to disk.

/workspace/{id}/undo

/workspace/{id}/redo

Undo or redo the most recent change of the workspace.

/workspace/{id}/go

Tell the current visible cue list of the given workspace to GO.

/workspace/{id}/stop

Stop playback but allow effects to continue. e.g., playback stops but echos continue.

/workspace/{id}/hardStop

Stop playback and cut all effects immediately.

/workspace/{id}/panic

A brief gradual fade out leading into a hard stop. A double panic will trigger an immediate hard stop.

/workspace/{id}/pause

/workspace/{id}/resume

/workspace/{id}/reset

/workspace/{id}/toggleFullScreen

/workspace/{id}/fullScreen {number}

Number is interpreted as a boolean, and sets whether the workspace is full screen. If no argument is provided, this returns the current full screen status of the workspace.

/workspace/{id}/toggleSelectionIsPlayhead

/workspace/{id}/selectionIsPlayhead {number}

Number is interpreted as a boolean, and sets whether the selection is the playhead. If no argument is provided, this returns whether the selection is currently locked to always be the playhead.

/workspace/{id}/toggleEditShowMode

/workspace/{id}/showMode {number}

Number is interpreted as a boolean, and sets whether the workspace is in show mode. If no argument is provided, this returns whether the workspace is currently in show mode.

/workspace/{id}/liveFadePreview {number}

Number is interpreted as a boolean, and sets whether the workspace has live fade previews enabled. If no argument is provided, this returns whether the workspace has live fade previews enabled.

/workspace/{id}/select/next

/workspace/{id}/select/previous

Select the next or previous cue.

/workspace/{id}/select/{cue_number}

/workspace/{id}/select_id/{id}

These methods will select the given cue(s).

/workspace/{id}/playhead/next

/workspace/{id}/playhead/previous

/workspace/{id}/playbackPosition/next

/workspace/{id}/playbackPosition/previous

Move the playhead (aka playbackPosition) to the the next or previous cue.

/workspace/{id}/playhead/nextSequence

/workspace/{id}/playhead/previousSequence

/workspace/{id}/playbackPosition/nextSequence

/workspace/{id}/playbackPosition/previousSequence

Move the playhead (aka playbackPosition) to the top of the next or previous cue sequence.

/workspace/{id}/playhead/{cue_number}

/workspace/{id}/playbackPosition/{cue_number}

These methods will set the playback position to the given cue.

/workspace/{id}/new cue_type

`cue_type` is a string describing which kind of cue to create. Supported strings include: `audio`, `mic`, `video`, `camera`, `fade`, `osc`, `midi`, `midi file`, `timecode`, `group`, `start`, `stop`, `pause`, `load`, `reset`, `devamp`, `goto`, `target`, `arm`, `disarm`, `wait`, `memo`, `script`, `cuelist`, or `cue list`.

This method returns the unique ID of the new cue. The newly created cue will also be selected, so subsequent commands can address the new cue either using the unique ID or simply addressing the currently selected cue.

/workspace/{id}/delete/{cue_number}

/workspace/{id}/delete/selected

/workspace/{id}/delete_id/{cue_id}

Delete the given cue(s).

/workspace/{id}/renumber startAtNumber incrementByNumber

Renumbers the selected cues.

/workspace/{id}/cueLists

/workspace/{id}/selectedCues

/workspace/{id}/runningCues

/workspace/{id}/runningOrPausedCues

All return an array of cue dictionaries:

```
[
    {
        "uniqueID": string,
        "number": string
        "name": string
        "listName": string
        "type": string
        "colorName": string
        "flagged": number
        "armed": number
    }
]
```

If the cue is a group, the dictionary will include an array of cue dictionaries for all children in the group:

```
[
    {
        "uniqueID": string,
        "number": string
        "name": string
        "listName": string
        "type": string
        "colorName": string
```

```
        "flagged": number
        "armed": number
        "cues": [ { }, { }, { } ]
    }
]
```

`colorName` may be: none, red, orange, green, blue, or purple.

Note: Methods that reply with an array of cue dictionaries may generate large OSC messages. These messages can easily grow larger than the maximum size supported by UDP datagrams. If you need to access these methods you should communicate to QLab over a TCP connection rather than a UDP connection.

/workspace/{id}/cue/{number}

/workspace/{id}/cue_id/{id}

These are addresses for a specific cue in a specific workspace. See below for cue commands that can be invoked on these addresses.

Cue methods

Cue methods can be invoked either on a specific workspace (using the `/workspace/{id}/cue/...` address pattern as described above) or “rootless”, where the `/cue/...` address pattern is applied to the current front-most workspace.

Cues can be addressed either by their cue number or their unique ID.

Cues always have a unique ID. They do not always have a number, but if it exists it will be unique within the workspace.

For the commands below, any instance of the address pattern:

```
/cue/{number}
```

Can be replaced by the equivalent unique ID address:

```
/cue_id/{id}
```

Alternately, the currently selected cue or cues in the front workspace can be addressed by the pattern:

```
/cue/selected
```

Or the current playback position can be addressed by either of the equivalent patterns:

```
/cue/playbackPosition
```

```
/cue/playhead
```

Finally, because QLab supports OSC address patterns, you may use OSC wildcards in {number} or {id} specifications. For example:

`/cue/*/armed 0` would disarm all cues in the workspace.

`/cue/understudy-*/colorName green` would set green as the display color for all cues that have a number that starts with the string “understudy-”.

Increment/Decrement Syntax

Simple number properties of cues can be incremented or decremented with the following syntax:

```
/cue/1/property/+ delta
```

```
/cue/1/property/- delta
```

For example, the command `/cue/10/preWait/+ 1` would increase the preWait of cue 10 by one second.

/cue/{number}/valuesForKeys json_string

This special method can be used to request a custom collection of state about the given cue. `json_string` must be a JSON-formatted string representing an array of keys you wish to query. For example:

```
/cue/2/valuesForKeys "[\"opacity\",\"surfaceSize\"]"
```

Would return the values of opacity and surfaceSize of video cue number 2.

/cue/{number}/valuesForKeysWithArguments json_string

This special method can be used to request a custom collection of state about the given cue. `json_string` must be a JSON-formatted string representing a dictionary of keys and arguments you wish to query. For example:

```
/cue/2/valuesForKeysWithArguments "{\"level\":[0,0]}"
```

Would return a dictionary that contains the value of the master volume level of a cue. Note that this method is limited to returning one value per key, even if you send multiple keys with different arguments.

/cue/{number}/start

/cue/{number}/stop

/cue/{number}/hardStop

/cue/{number}/pause

/cue/{number}/resume

/cue/{number}/togglePause

/cue/{number}/load

/cue/{number}/preview

/cue/{number}/reset

/cue/{number}/panic

Tell the specified cue to start/stop/hardStop/pause/resume/togglePause/load/preview/reset/panic.

/cue/{number}/loadAt {number}

If given a number, load the cue at that time in seconds. If no number, equivalent to `load`.

/cue/{number}/uniqueID

/cue/{number}/type

/cue/{number}/defaultName

/cue/{number}/displayName

/cue/{number}/listName

/cue/{number}/hasFileTargets

/cue/{number}/hasCueTargets

/cue/{number}/allowsEditingDuration

/cue/{number}/isLoaded

/cue/{number}/isRunning

/cue/{number}/isPaused

/cue/{number}/isBroken

/cue/{number}/preWaitElapsed

/cue/{number}/actionElapsed

/cue/{number}/postWaitElapsed

/cue/{number}/percentPreWaitElapsed

/cue/{number}/percentActionElapsed

/cue/{number}/percentPostWaitElapsed

Read-only properties.

/cue/{number}/number {string}

/cue/{number}/name {string}

/cue/{number}/notes {string}

Getters/setters for the given property. Give an argument to set the property.

/cue/{number}/fileTarget {string}

Getter/setter for the file target, if the cue has file targets. You can provide three kinds of paths:

- Full paths, e.g. `/a/full/path/to/some/file.wav`
- Paths beginning with a tilde, e.g. `~/a/path/to some/file.mov`
- Relative paths, e.g. `this/is/a/relative/path.mid`

Paths beginning with a tilde (~) will be expanded; the tilde signifies "relative to the user's home directory".

Relative paths will be interpreted according to the current working directory. Use QLab's `/workingDirectory` method to set or get the current working directory.

/cue/{number}/cueTargetNumber {string}

/cue/{number}/cueTargetId {string}

/cue/{number}/preWait {number}

/cue/{number}/duration {number}

/cue/{number}/postWait {number}

Getters/setters for the given property. Give an argument to set the property.

/cue/{number}/continueMode {number}

Get/set the continue mode. Values can be:

```
0 - NO CONTINUE
1 - AUTO CONTINUE
2 - AUTO FOLLOW
```

/cue/{number}/flagged {number}

/cue/{number}/autoLoad {number}

/cue/{number}/armed {number}

The final {number} is interpreted as a boolean; 0 equals false, any other number equals true.

/cue/{number}/colorName {string}

Getters/setters for the given property. Give an argument to set the property.

/cue/{number}/children

Read-only; gets a list of the children of the cue. Returns the same data as the workspace /cueLists method.

Group cue methods

Methods specific to group cues.

/cue/{number}/mode {number}

Getter/setter for the mode of the group cue. Valid modes to set on a group are 1, 2, 3, or 4. A cue list will return mode 0, but the mode can not be set to 0.

/cue/{number}/playheadId {string}

/cue/{number}/playbackPositionId {string}

Getter/setter for the playback position of the cue.

As a getter, returns a cue uniqueID, or “none” if there is no current playback position.

As a setter, provide a cue uniqueID `string`.

Only applies if the group is a cue list.

/cue/{number}/playheadNumber {string}

/cue/{number}/playbackPositionNumber {string}

Getter/setter for the playback position of the cue.

As a getter, returns a cue number (which may be an empty string), or “none” if there is no current playback position.

As a setter, provide a cue number `string`.

Only applies if the group is a cue list.

Audio cue methods

Methods specific to audio cues.

/cue/{number}/patchList

Read-only; return a list of audio patches defined for this workspace:

```
[
    {
        "patchNumber": integer,
        "patchName": string
    }
]
```

/cue/{number}/patch {number}

A number from 1 to 8.

/cue/{number}/startTime {number}

/cue/{number}/endTime {number}

Times must be entered in seconds as any whole or decimal number greater than or equal to zero.

/cue/{number}/playCount {number}

playCount can be any whole number greater than or equal to 0.

/cue/{number}/infiniteLoop {number}

The final {number} is interpreted as a boolean; 0 equals false, any other number equals true.

/cue/{number}/rate {number}

Rate can be any positive decimal value from 0.03 to 33.0.

/cue/{number}/doPitchShift {number}

The final {number} is interpreted as a boolean; 0 equals false, any other number equals true.

/cue/{number}/doFade {number}

Getters/setters for the given property. Give an argument to set the property.

/cue/{number}/sliderLevel channel {decibel}

/cue/{number}/sliderLevel/channel {decibel}

Get or set a single slider volume level.

`channel` is either an integer from 0 to 48, or a string (the channel name). 0 is the master slider.

`decibel` is an optional floating point number. When present it is the decibel value to set.

If `decibel` is sent as a string (e.g. "-inf") QLab will use the minimum decibel value set in the workspace settings.

/cue/{number}/sliderLevels

Read-only; returns an array of 49 numbers.

/cue/{number}/level inChannel outChannel {decibel}

/cue/{number}/level/inChannel/outChannel {decibel}

Get or set a single volume level.

`inChannel` is an integer from 0 to 24.

`outChannel` is either an integer from 0 to 48, or a string (the channel name).

`decibel` is an optional floating point number. When present it is the decibel value to set.

If `decibel` is sent as a string (e.g. "-inf") QLab will use the minimum decibel value set in the workspace settings.

/cue/{number}/level/inChannel/outChannel/+ decibel

/cue/{number}/level/inChannel/outChannel/- decibel

Similar to the above method, but makes a relative adjustment adding or subtracting the given decibels to the given level entry.

/cue/{number}/gang inChannel outChannel {gang}

/cue/{number}/gang/inChannel/outChannel {gang}

Get or set a single level gang.

`inChannel` is an integer from 0 to 24.

`outChannel` is either an integer from 0 to 48, or a string (the channel name).

`gang` is an optional text string. When present it is the gang to set.

/cue/{number}/liveRate {number}

These "live" methods are the same as their non-live counterparts, but operate on the active, "live" value of a running cue, rather than changing the "start state" of the cue. Invoking these methods does not cause the document to have unsaved changes.

Mic cue methods

Methods specific to microphone cues.

/cue/{number}/patchList

Read-only; return a list of audio patches defined for this workspace:

```
[
    {
        "patchNumber": integer,
        "patchName": string
    }
]
```

/cue/{number}/patch {number}

A number from 1 to 8.

/cue/{number}/sliderLevel channel {decibel}

/cue/{number}/sliderLevel/channel {decibel}

Get or set a single slider volume level.

`channel` is either an integer from 0 to 48, or a string (the channel name). 0 is the master slider.

`decibel` is an optional floating point number. When present it is the decibel value to set.

If `decibel` is sent as a string (e.g. “-inf”) QLab will use the minimum decibel value set in the workspace settings.

/cue/{number}/sliderLevels

Read-only; returns an array of 49 numbers.

/cue/{number}/level inChannel outChannel {decibel}

/cue/{number}/level/inChannel/outChannel {decibel}

Get or set a single volume level.

`inChannel` is an integer from 0 to 24.

`outChannel` is either an integer from 0 to 48, or a string (the channel name).

`decibel` is an optional floating point number. When present it is the decibel value to set.

If `decibel` is sent as a string (e.g. “-inf”) QLab will use the minimum decibel value set in the workspace settings.

/cue/{number}/level/inChannel/outChannel/+ decibel

/cue/{number}/level/inChannel/outChannel/- decibel

Similar to the above method, but makes a relative adjustment adding or subtracting the given decibels to the given level entry.

/cue/{number}/gang inChannel outChannel {gang}

/cue/{number}/gang/inChannel/outChannel {gang}

Get or set a single level gang.

`inChannel` is an integer from 0 to 24.

`outChannel` is either an integer from 0 to 48, or a string (the channel name).

`gang` is an optional text string. When present it is the gang to set.

Video cue methods

Methods specific to video cues.

/cue/{number}/surfaceList

Read-only; return a list of surfaces defined for this workspace:

```
[
    {
        "surfaceName": string,
        "surfaceID": number
    }
]
```

/cue/{number}/surfaceID {number}

Get/set the surface ID of the cue.

/cue/{number}/patchList

Read-only; return a list of audio patches defined for this workspace:

```
[
    {
        "patchNumber": integer,
        "patchName": string
    }
]
```

/cue/{number}/patch {number}

Get/set the audio patch of the cue. A number from 1 to 8.

/cue/{number}/startTime {number}

/cue/{number}/endTime {number}

Times must be entered in seconds as any whole or decimal number greater than or equal to zero.

/cue/{number}/playCount {number}

playCount can be any whole number greater than or equal to 0.

/cue/{number}/infiniteLoop {number}

The final {number} is interpreted as a boolean; 0 equals false, any other number equals true.

/cue/{number}/rate {number}

Rate can be any positive decimal value from 0.03 to 33.0.

/cue/{number}/doPitchShift {number}

The final {number} is interpreted as a boolean; 0 equals false, any other number equals true.

/cue/{number}/holdLastFrame {number}

/cue/{number}/doFade {number}

/cue/{number}/fullScreen {number}

Getters/setters for the given property. Give an argument to set the property.

/cue/{number}/layer {number}

A number between 0 and 1000, inclusive.

/cue/{number}/opacity {number}

A float between 0 and 1.

/cue/{number}/preserveAspectRatio {number}

The final {number} is interpreted as a boolean; 0 equals false, any other number equals true.

/cue/{number}/translationX {number}

/cue/{number}/translationY {number}

/cue/{number}/scaleX {number}

/cue/{number}/scaleY {number}

/cue/{number}/originX {number}

/cue/{number}/originY {number}

Getters/setters for the given property; translation, scale, and origin (anchor). Give an argument to set the property.

/cue/{number}/quaternion {number number number number}

Get/set an array of four floats representing the video's orientation as a quaternion.

/cue/{number}/cueSize

Read-only; returns the natural size of the cue's video frame:

```
{
    "width": number,
    "height": number
}
```

/cue/{number}/surfaceSize

Read-only; returns the size of the cue's display surface:

```
{
    "width": number,
    "height": number
}
```

/cue/{number}/doEffect {number}

Get/set whether video processing is enabled. The {number} parameter is interpreted as a boolean.

/cue/{number}/effect {number}

Get/set which effect is used:

/cue/{number}/effectSet keyString value

Set the Quartz Composer input for keyString to the given value. Value can be a string or number.

/cue/{number}/sliderLevel channel {decibel}

/cue/{number}/sliderLevel/channel {decibel}

Get or set a single slider volume level.

`channel` is either an integer from 0 to 48, or a string (the channel name). 0 is the master slider.

`decibel` is an optional floating point number. When present it is the decibel value to set.

If `decibel` is sent as a string (e.g. "-inf") QLab will use the minimum decibel value set in the workspace settings.

/cue/{number}/sliderLevels

Read-only; returns an array of 49 numbers.

/cue/{number}/level inChannel outChannel {decibel}

/cue/{number}/level/inChannel/outChannel {decibel}

Get or set a single volume level.

`inChannel` is an integer from 0 to 24.

`outChannel` is either an integer from 0 to 48, or a string (the channel name).

`decibel` is an optional floating point number. When present it is the decibel value to set.

If `decibel` is sent as a string (e.g. "-inf") QLab will use the minimum decibel value set in the workspace settings.

/cue/{number}/level/inChannel/outChannel/+ decibel

/cue/{number}/level/inChannel/outChannel/- decibel

Similar to the above method, but makes a relative adjustment adding or subtracting the given decibels to the given level entry.

/cue/{number}/gang inChannel outChannel {gang}

/cue/{number}/gang/inChannel/outChannel {gang}

Get or set a single level gang.

`inChannel` is an integer from 0 to 24.

`outChannel` is either an integer from 0 to 48, or a string (the channel name).

`gang` is an optional text string. When present it is the gang to set.

/cue/{number}/liveRate {number}

/cue/{number}/liveEffectSet keyString value

These "live" methods are the same as their non-live counterparts, but operate on the active, "live" value of a running cue, rather than changing the "start state" of the

cue. Invoking these methods does not cause the document to have unsaved changes.

Titles cue methods

In addition to the following, titles cues respond to all the same methods as video cues.

/cue/{number}/text {string}

Get/set the text of a titles cue as an unformatted string. When setting, the formatting will match the first character of the existing text.

/cue/{number}/liveText {string}

Get/set the live text of a titles cue as an unformatted string. When setting, the formatting will match the first character of the existing text. The live text will be the same as the text by default. If you set the live text, it will change what the cue displays without marking the workspace as edited. Resetting the cue will change the text back to the initial text.

/cue/{number}/fixedWidth {number}

Get/set the “fixedWidth” property of a Titles cue.

Fade cue methods

Methods specific to fade cues.

/cue/{number}/sliderLevel channel {decibel}

/cue/{number}/sliderLevel/channel {decibel}

Get or set a single slider volume level.

`channel` is an integer from 0 to 48. 0 is the master slider.

`decibel` is an optional floating point number. When present it is the decibel value to set.

If `decibel` is sent as a string (e.g. "-inf") QLab will use the minimum decibel value set in the workspace settings.

/cue/{number}/sliderLevels

Read-only; returns an array of 49 numbers.

/cue/{number}/level inChannel outChannel {decibel}

/cue/{number}/level/inChannel/outChannel {decibel}

Get or set a single volume level.

`inChannel` is an integer from 0 to 24.

`outChannel` is an integer from 0 to 48.

`decibel` is an optional floating point number. When present it is the decibel value to set.

If `decibel` is sent as a string (e.g. "-inf") QLab will use the minimum decibel value set in the workspace settings.

/cue/{number}/level/inChannel/outChannel/+ decibel

/cue/{number}/level/inChannel/outChannel/- decibel

Similar to the above method, but makes a relative adjustment adding or subtracting the given decibels to the given level entry.

/cue/{number}/gang inChannel outChannel {gang}

/cue/{number}/gang/inChannel/outChannel {gang}

Get or set a single level gang.

`inChannel` is an integer from 0 to 24.

`outChannel` is an integer from 0 to 48.

`gang` is an optional text string. When present it is the gang to set.

OSC cue methods

Methods specific to OSC cues.

/cue/{number}/patch {number}

A number from 1 to 16.

/cue/{number}/messageType {number}

/cue/{number}/qlabCommand {number}

/cue/{number}/qlabCueNumber {string}

/cue/{number}/qlabCueParameters {string}

/cue/{number}/customString {string}

/cue/{number}/udpString {string}

MIDI cue methods

Methods specific to MIDI cues.

/cue/{number}/duration {number}

If the cue fades over a value, this is the duration of the fade.

/cue/{number}/patch {number}

A number from 1 to 8.

/cue/{number}/messageType {number}

```
1 = MIDI Voice Message ("Musical MIDI")
2 = MIDI Show Control Message (MSC)
3 = MIDI SysEx Message
```

/cue/{number}/status {number}

```
0 = Note Off
1 = Note On
2 = Key Pressure (Aftertouch)
3 = Control Change
4 = Program Change
```

```
5 = Channel Pressure Change
6 = Pitch Bend Change
```

/cue/{number}/channel {number}

A number from 1 to 16.

/cue/{number}/byte1 {number}

A number from 0 to 127.

/cue/{number}/byte2 {number}

A number from 0 to 127.

/cue/{number}/byteCombo {number}

A number from 0 to 16383.

/cue/{number}/doFade {number}

A boolean.

/cue/{number}/endValue {number}

A number from 0 to 127, unless fading a pitch bend, in which case it can go to 16383.

/cue/{number}/deviceID {number}

A number from 0 to 127.

/cue/{number}/commandFormat {number}

A number from 1 to 127, but only the following values are meaningful:

```
127 = All Types
1   = Lighting (General)
2   = Moving Lights
3   = Color Changers
4   = Strobes
5   = Lasers
6   = Chasers
16  = Sound (General)
17  = Music
18  = CD Players
19  = EPROM Playback
20  = Audio Tape Machines
21  = Intercoms
22  = Amplifiers
23  = Audio Effects Devices
24  = Equalizers
32  = Machinery (General)
33  = Rigging
34  = Flys
35  = Lifts
36  = Turntables
37  = Trusses
38  = Robots
39  = Animation
40  = Floats
41  = Breakaways
42  = Barges
48  = Video (General)
49  = Video Tape Machines
50  = Video Cassette Machines
51  = Video Disc Players
52  = Video Switchers
53  = Video Effects
54  = Video Character Generators
55  = Video Still Stores
56  = Video Monitors
```

```
64  = Projection (General)
65  = Film Projectors
66  = Slide Projectors
67  = Video Projectors
68  = Dissolvers
69  = Shutter Controls
80  = Process Control (General)
81  = Hydraulic Oil
82  = H2O
83  = CO2
84  = Compressed Air
85  = Natural Gas
86  = Fog
87  = Smoke
88  = Cracked Haze
96  = Pyrotechnics (General)
97  = Fireworks
98  = Explosions
99  = Flame
100 = Smoke Pots
```

/cue/{number}/command {number}

A number from 1 to 127, but only the following values are meaningful:

```
1  = GO
2  = STOP
3  = RESUME
4  = TIMED_GO
5  = LOAD
6  = SET
7  = FIRE
8  = ALL_OFF
9  = RESTORE
10 = RESET
```

```
11 = GO_OFF
16 = GO/JAM_CLOCK
17 = STANDBY_+
18 = STANDBY_-
19 = SEQUENCE_+
20 = SEQUENCE_-
21 = START_CLOCK
22 = STOP_CLOCK
23 = ZERO_CLOCK
24 = SET_CLOCK
25 = MTC_CHASE_ON
26 = MTC_CHASE_OFF
27 = OPEN_CUE_LIST
28 = CLOSE_CUE_LIST
29 = OPEN_CUE_PATH
30 = CLOSE_CUE_PATH
```

/cue/{number}/qNumber {string}

/cue/{number}/qList {string}

/cue/{number}/qPath {string}

/cue/{number}/macro {number}

A number from 0 to 127.

/cue/{number}/controlNumber {number}

A number from 0 to 16383.

/cue/{number}/controlValue {number}

A number from 0 to 16383.

/cue/{number}/timecodeString {string}

/cue/{number}/hours {number}

/cue/{number}/minutes {number}

/cue/{number}/seconds {number}

/cue/{number}/frames {number}

/cue/{number}/subframes {number}

/cue/{number}/timecodeFormat {number}

/cue/{number}/rawString {string}

MIDI file cue methods

Methods specific to MIDI File cues.

/cue/{number}/patch {number}

/cue/{number}/rate {number}

Devamp cue methods

Methods specific to Devamp cues.

/cue/{number}/startNextCueWhenSliceEnds {number}

/cue/{number}/stopTargetWhenSliceEnds {number}

Booleans.

Script cue methods

Methods specific to Script cues.

/cue/{number}/scriptSource

Read only; returns a string representing the AppleScript contained in the cue.

AppleScript Dictionary

The list of commands, functions, and properties that AppleScript can use to interact with an application is called that application's dictionary. You can find QLab's AppleScript dictionary here, or view it within the Script Editor application, which is found in /Applications/Utilities.

In Script Editor, choose *Open Dictionary...* from the **File** menu, and choose QLab from the list of applications.

The dictionary is grouped by "suite"; all applications that use AppleScript must include the *Standard Suite,* and then any application-specific commands or properties are generally grouped together into another suite named after the application.

continue modes

enum : Continue mode of a cue.

- **do_not_continue** : Do not automatically continue to the next cue.
- **auto_continue** : Automatically continue to the next cue after completing the post wait.
- **auto_follow** : Automatically continue to the next cue after completing the action of the cue.

group modes

enum

- **cue_list** : The group is a cue list.
- **fire_first_enter_group** : Fire first child and enter into group.
- **fire_first_go_to_next_cue** : Fire first child and go to next cue.
- **fire_all** : Fire all children simultaneously.
- **fire_random** : Fire a random child cue and then go to the next cue.

smpte format

enum : SMPTE timecode format.

- **fps_24** : 24 frames per second
- **fps_25** : 25 frames per second
- **fps_30_drop** : 30 frames per second (drop frame)
- **fps_30_non_drop** : 30 frames per second (non-drop)

osc type

enum

- **qlab** : QLab OSC message.
- **custom** : Custom OSC message.
- **udp** : Raw UDP message.

mtc ltc

enum

- **mtc** : MIDI Timecode
- **ltc** : Linear / Logitudinal Timecode

midi type

enum

- **voice** : MIDI Voice message (“Musical MIDI”).
- **msc** : MIDI Show Control message.
- **sysex** : MIDI Sysex message.

midi command

enum

- **note_on** : Note on.
- **note_off** : Note off.
- **program_change** : Program change.
- **control_change** : Control change.
- **key_pressure** : Key pressure (aftertouch).
- **channel_pressure** : Channel pressure.
- **pitch_bend** : Pitch bend (pitch wheel).

workspace

n [inh. document] : A QLab workspace.

Elements

- contains cue lists, cues.

Properties

- **unique id** (text, r/o) : The unique ID of the workspace.
- **current cue list** (cue list) : The currently visible cue list for the workspace.
- **selected** (list of cue) : The currently selected cue(s) in the currently visible cue list.
- **active cues** (list of cue, r/o) : The list of active cues (running or paused) in this workspace.
- **edit mode** (boolean) : Is the workspace currently in edit mode?
- **show mode** (boolean) : Is the workspace currently in show mode?
- **live fade preview** (enabled, disabled) : Live Fade Preview

Responds to

make, load, go, start, pause, stop, hardStop, reset, panic, moveSelectionUp, moveSelectionDown.

Inherited from document:

Elements

- contained by application.

Properties

- **path** (text) : The document's path.
- **modified** (boolean, r/o) : Has the document been modified since the last save?
- **name** (text) : The document's name.

Responds to

- close, print, save.

cue

n : A cue.

Elements

- contains cues; contained by workspaces, cues.

Properties

- **uniqueID** (text, r/o) : The unique ID of the cue.
- **parent** (cue, r/o) : The parent cue of this cue.
- **q type** (text, r/o) : The name of this kind of cue, e.g. “Audio”, “Video”, “MIDI”, etc.
- **q number** (text) : The number of the cue. Unique if present.
- **q name** (text) : The name of the cue. Not unique.
- **q list name** (text, r/o) : The name of the cue as displayed in the cue list. (i.e. Might be a default name.)
- **q display name** (text, r/o) : The name of the cue as displayed in the standby view. (i.e. Never empty.)
- **q default name** (text, r/o) : The name QLab would give the cue by default, if any.
- **notes** (text) : The notes for this cue.
- **cue target** (cue) : The cue this cue targets, if any.
- **file target** (file) : The file this cue targets, if any.
- **pre wait** (real) : The time in seconds before the action is triggered.
- **duration** (real) : The duration of the cue’s action in seconds. Not editable for all cue types.
- **post wait** (real) : The time in seconds until continuing on to the next cue.
- **continue mode** (do_not_continue, auto_continue, auto_follow) : Continue mode of the cue.
- **flagged** (boolean) : Is this cue flagged?
- **autoload** (boolean) : Does this cue auto load?
- **armed** (boolean) : Is this cue armed?
- **hotkey trigger** (enabled, disabled) : State of the HotKey trigger.

- **midi trigger** (enabled, disabled) : State of the MIDI trigger.
- **midi command** (note_on, note_off, program_change, control_change, key_pressure, channel_pressure, pitch_bend) : Type of MIDI command that will trigger the cue. (NOTE: pitch_bend messages are NOT accepted as remote MIDI triggers.)
- **midi byte one** (integer) : Byte 1 of the MIDI trigger.
- **midi byte two** (integer) : Byte 2 of the MIDI trigger.
- **midi byte one string** (text) : Display String of Byte 1 of the MIDI trigger.
- **midi byte two string** (text) : Display String of Byte 2 of the MIDI trigger.
- **timecode trigger** (enabled, disabled) : State of the timecode trigger.
- **wall clock trigger** (enabled, disabled) : State of the wall clock trigger.
- **wall clock hours** (integer) : Hours field of the wall clock trigger.
- **wall clock minutes** (integer) : Minutes field of the wall clock trigger.
- **wall clock seconds** (integer) : Seconds field of the wall clock trigger.
- **loaded** (boolean, r/o) : Is this cue loaded?
- **running** (boolean, r/o) : Is this cue running?
- **paused** (boolean, r/o) : Is this cue paused?
- **broken** (boolean, r/o) : Is this cue broken?
- **pre wait elapsed** (real, r/o) : The time in seconds that have elapsed on the pre wait.
- **action elapsed** (real, r/o) : The time in seconds that have elapsed in the action of the cue.
- **post wait elapsed** (real, r/o) : The time in seconds that have elapsed on the post wait.

- **percent pre wait elapsed** (real, r/o) : The percent of the pre wait that has elapsed.
- **percent action elapsed** (real, r/o) : The percent of the cue's action that has elapsed.
- **percent post wait elapsed** (real, r/o) : The percent of the post wait that has elapsed.

Responds to

- load, preview, start, pause, stop, hardStop, reset, panic.

group cue

n [inh. cue] : A Group Cue.

Properties

- **mode** (cue_list, fire_first_enter_group, fire_first_go_to_next_cue, fire_all, fire_random) : The firing style of this group.

cue list

n [inh. group cue > cue] : A cue list.

Elements

- contained by workspaces.

Properties

- **playback position** (cue) : The playback position of this cue list is the cue that will fire at the next GO.
- **sync to timecode** (enabled, disabled) : Sync the cues in this cue list to incoming timecode.
- **sync mode** (mtc, ltc) : Which kind of incoming timecode this cue list listens for.
- **smpte format** (fps_24, fps_25, fps_30_drop, fps_30_non_drop) : SMPTE format of the incoming timecode.
- **mtc sync source name** (text) : Name of the MIDI device which feeds us MTC timecode.
- **ltc sync channel** (integer) : Audio channel that carries the LTC signal.

audio cue

n [inh. cue] : An Audio Cue.

Properties

- **patch** (integer) : Audio device patch number.
- **start time** (real) : Time in the file where playback begins.
- **end time** (real) : Time in the file where playback ends.
- **play count** (integer) : Number of times the audio between the start and end times plays. Always >= 1.
- **infinite loop** (boolean) : Does the cue loop infinitely?
- **rate** (real) : Playback rate of the audio cue.
- **integrated fade** (enabled, disabled) : State of the integrated fade - enabled or disabled.
- **audio input channels** (integer, r/o) : The number of audio input channels for the cue. (i.e. The number of distinct channels in the file.)
- **pitch shift** (enabled, disabled) : Whether pitch shifting is enabled or disabled.

Responds to

- getLevel, setLevel, getGang, setGang.

mic cue

n [inh. cue] : A Microphone Cue.

Properties

patch (integer) : Audio device patch number. audio input channels (integer, r/o) : The number of audio input channels for the cue.

Responds to

getLevel, setLevel, getGang, setGang.

video cue

n [inh. cue] : A Video Cue.

Properties

- **patch** (integer) : Audio device patch number.
- **start time** (real) : Time in the file where playback begins.
- **end time** (real) : Time in the file where playback ends.
- **infinite loop** (boolean) : Does the cue loop infinitely?
- **rate** (real) : Playback rate of the video cue.
- **layer** (integer) : Display layer of the video.
- **full screen** (boolean) : Is the cue displaying in full screen mode?
- **preserve aspect ratio** (boolean) : Does the cue preserve aspect ratio?
- **opacity** (real) : Video opacity.
- **translation x** (real) : Translation along the x axis.
- **translation y** (real) : Translation along the y axis.
- **scale x** (real) : Scale along the x axis.
- **scale y** (real) : Scale along the y axis.
- **do video effect** (boolean) : Apply a video effect?
- **custom quartz file** (file) : The custom Quartz Composer file used to render this cue.
- **audio input channels** (integer, r/o) : The number of audio input channels for the cue. (i.e. The number of distinct channels in the file.)
- **pitch shift** (enabled, disabled) : Whether pitch shifting is enabled or disabled.
- **hold at end** (boolean) : Should the final frame of the video be left visible when playback reaches the end of the file?

Responds to

- getLevel, setLevel, getGang, setGang.

titles cue

n [inh. video cue > cue] : A Titles Cue.

Properties

- **text** (text) : Text of the titles cue.
- **live text** (text) : Live text of the titles cue. Setting this does not mark the workspace as edited.

camera cue

n [inh. cue] : A Camera Cue.

Properties

- **camera patch** (integer) : Camera patch number.
- **layer** (integer) : Display layer of the video.
- **full screen** (boolean) : Is the cue displaying in full screen mode?
- **preserve aspect ratio** (boolean) : Does the cue preserve aspect ratio?
- **opacity** (real) : Video opacity.
- **translation x** (real) : Translation along the x axis.
- **translation y** (real) : Translation along the y axis.
- **scale x** (real) : Scale along the x axis.
- **scale y** (real) : Scale along the y axis.
- **do video effect** (boolean) : Apply a video effect?
- **custom quartz file** (file) : The custom Quartz Composer file used to render this cue.

fade cue

n [inh. cue] : A Fade Cue.

Properties

- **fade mode** (absolute, relative) : Absolute or relative mode.
- **stop target when done** (boolean) : Stop the target when this cue completes?
- **opacity** (real) : Video opacity.
- **translation x** (real) : Translation along the x axis.
- **translation y** (real) : Translation along the y axis.
- **scale x** (real) : Scale along the x axis.
- **scale y** (real) : Scale along the y axis.
- **preserve aspect ratio** (boolean) : Does the cue preserve aspect ratio?
- **rotation type** (integer) : Quaternion, x-axis, y-axis, or z-axis.
- **rotation** (real) : Rotation in degrees.
- **do opacity** (boolean) : Does the cue animate opacity?
- **do translation** (boolean) : Does the cue animate translation?
- **do scale** (boolean) : Does the cue animate scale?
- **do rotation** (boolean) : Does the cue animate rotation?

Responds to

- getLevel, setLevel, getGang, setGang.

midi cue

n [inh. cue] : A MIDI Cue.

Properties

- **patch** (integer) : MIDI device patch number.
- **message type** (voice, msc, sysex) : The type of MIDI message.
- **command** (note_on, note_off, program_change, control_change, key_pressure, channel_pressure, pitch_bend) : The MIDI command.
- **channel** (integer) : MIDI channel number.
- **byte one** (integer) : First byte of the message.
- **byte two** (integer) : Second byte of the message.
- **byte combo** (integer) : Value when first and second bytes are interpreted as parts of one number. Used for pitch bend messages.
- **start value** (integer, r, o) : The start value for the MIDI fade.
- **end value** (integer) : The end value for the MIDI fade.
- **fade** (enabled, disabled) : State of the MIDI fade.
- **deviceID** (integer) : MIDI Show Control device ID.
- **command format** (integer) : MIDI Show Control command format.
- **command number** (integer) : MIDI Show Control command.
- **q_number** (text) : Q Number message parameter.
- **q_list** (text) : Q List message parameter.
- **q_path** (text) : Q Path message parameter.
- **macro** (integer) : MSC macro.
- **control number** (integer) : MSC control number.
- **control value** (integer) : MSC control value.
- **hours** (integer) : MSC hours parameter.
- **minutes** (integer) : MSC minutes parameter.
- **seconds** (integer) : MSC seconds parameter.

- **frames** (integer) : MSC frames parameter.
- **subframes** (integer) : MSC subframes parameter.
- **smpte format** (fps_24, fps_25, fps_30_drop, fps_30_non_drop) : SMPTE format of the timecode parameters.
- **send time with set** (boolean) : Send the timecode parameters with the SET command?
- **sysex message** (text) : The raw SysEx message. Use only hexadecimal characters and whitespace. Omit the starting F0 and the ending F7.

midi file cue

n [inh. cue] : A MIDI File Cue.

Properties

- **patch** (integer) : MIDI device patch number.
- **rate** (real) : Playback rate of the MIDI File cue.

osc cue

n [inh. cue] : An OSC Cue.

Properties

- **patch** (integer) : OSC destination patch number.
- **osc message type** (qlab, custom, udp) : The type of OSC message.
- **q_num** (text) : The QLab cue number, for QLab type messages.
- **q_command** (number) : The QLab OSC command, for QLab type messages.
- **q_params** (text) : The QLab command parameters, for QLab type messages. Not all messages have parameters.
- **custom message** (text) : The custom OSC message, for custom type messages.
- **udp message** (text) : The raw UDP message, for udp type messages.

timecode cue

n [inh. cue] : A Timecode Cue to generate MTC or LTC timecode.

Properties

- **patch** (integer) : MIDI destination patch number.
- **smpte format** (fps_24, fps_25, fps_30_drop, fps_30_non_drop) : SMPTE format of the outgoing timecode.
- **start time offset** (real) : Time in seconds where the MTC clock begins counting.

devamp cue

n [inh. cue] : A Devamp Cue.

Properties

- **fire next cue when slice ends** (boolean) : Fire the next cue at the moment the target slice ends?
- **stop target when slice ends** (boolean) : Stop the target at the moment the target slice ends?

load cue

n [inh. cue] : A Load Cue.

Properties

- **load time** (real) : Load target cue to this time.

target cue

n [inh. cue] : A Target Cue.

Properties

- **assigned number** (text) : Number of cue to assign. The cue with this number will be assigned as the new target.

script cue

n [inh. cue] : A Script Cue.

Properties

- **script source** (text) : AppleScript source for the cue. The script will be recompiled when set.

Responds to

- compile.

make

v

make workspace
type text : Name of the kind of cue you want to make. (Audio, Video, Camera, MIDI, etc.) Pass “cue list” to make a new cue list.

load

v : Load a cue or workspace to a given time.

load specifier : The cue(s) or workspace(s) to load.
[**time** real] : Load time.

go

v : Make a workspace GO.

go specifier : The workspace to GO.

preview

v : Preview one or more cues.

Previewing starts only the action of the cue, skipping any pre-wait and not continuing to other cues.
preview specifier : The cue(s) to preview.

start

v : Start one or more cues or workspaces.

start specifier : The cue(s) or workspace(s) to start. For a workspace, “start” means to unpause all paused cues.

pause

v : Pause one or more cues or workspaces.

pause specifier : The cue(s) or workspace(s) to pause.

stop

v : Stop one or more cues or workspaces.

stop specifier : The cue(s) or workspace(s) to stop.

hardStop

v : Hard stop one or more cues or workspaces.

hardStop specifier : The cue(s) or workspace(s) to stop.

panic

v : Panic one or more cues or workspaces.

panic specifier : The cue(s) or workspace(s) to panic.

reset

v : Reset one or more cues or workspaces.

reset specifier : The cue(s) or workspace(s) to reset.

moveSelectionUp

v : Select the previous cue.

moveSelectionUp specifier : The workspace whose selection will change.

moveSelectionDown

v : Select the next cue.

moveSelectionDown specifier : The workspace whose selection will change.

getLevel

v

getLevel cue : The cue for which you want to get the volume level.
row integer : The row of the level matrix. Row 0 = the master and output levels.
column integer : The column of the level matrix. Column 0 = the master and input levels.
returns real : The value in decibels of the level at the specified location in the matrix.

setLevel

v

setLevel cue : The cue for which you want to adjust the volume level.
row integer : The row of the level matrix. Row 0 = the master and output levels.
column integer : The column of the level matrix. Column 0 = the master and input levels.
db real : The decibel value for the volume level.

getGang

v

getGang cue : The cue for which you want to adjust the gang.
row integer : The row of the level matrix. Row 0 = the master and output levels.
column integer : The column of the level matrix. Column 0 = the master and input levels.
returns text : The value of the gang at the specified location in the matrix.

setGang

v

setGang cue : The cue for which you want to adjust the gang.
row integer : The row of the level matrix. Row 0 = the master and output levels.
column integer : The column of the level matrix. Column 0 = the master and input levels.
gang text : The gang value to set.

compile

v

compile script cue : The Script cue whose source you want to recompile.

Other Resources

The Figure 53 Wiki

You can find the QLab Wiki at wiki.figure53.com. There, you can find a large and ever-changing repository of user-contributed information about QLab, include a good number of example AppleScript-related tips and examples.

Note that many scripts which were originally written for QLab 2 may not work in QLab 3 without adjustments.

The QLab Cook Book

The QLab Cook Book, which you can find at http://qlabcookbook.com/, is the creation of Mic Pool, a wonderful and talented individual. The Cook Book contains a number of advanced and often complex sample workspaces, many of which include elaborate Script cues. Mic has generously published these workspaces for one and all to download and to learn from.

Chapter 7: How To

Crossfade Sounds

For this How-To, let's suppose that you want to play some scene change music, and then when the scene change completes, you want to crossfade to the background atmosphere of a city park over the course of ten seconds.

There is more than one way to make a crossfade, but this way is particularly tidy.

1. Create an Audio cue for your scene change music. Name the cue “Music.”
2. Set the levels of that cue to suit your taste and needs.
3. Create a Fade cue that targets “Music” and sets its level to `-inf`, with the *Stop target when done* box checked. Set the duration of this Fade cue to 10 seconds.
4. Create an Audio cue for your city park atmosphere. Name it “Park.”
5. Set the levels of that cue to the levels at which you want it to play during the scene, after the crossfade happens.
6. With the “Park” cue selected, choose *Copy Levels* from the **Tools** menu.
7. Create a Fade cue that targets “Park,” and with this Fade cue selected choose *Paste Levels* from the **Tools** menu. Set the duration of this Fade cue to 10 seconds.
8. In the Levels tab of the inspector, click once on the master level text field to turn it yellow.
9. Select the “Park” cue, and bring its master level to `-inf`
10. Create a Group cue and name it “Crossfade to park.”
11. Set the mode of the Group cue to “Start all children simultaneously.”
12. Place the “Park” cue and both Fade cues within the Group cue.
13. Trigger the “Music” cue. When you’re ready, trigger the “Crossfade to park” cue.
14. Voila!

Use Reverb with Mic Cues

This How-To is quite wordy, because this concept is a little complicated. However, by going step by step, the complexity should unravel somewhat.

The trouble with reverb on Mic cues, in general, is that most reverb plug-ins only work with a specific number of inputs (generally one or two), and that is seldom the number of input channels available in a Mic cue.

The reason for this is that the number of inputs on a Mic cue is equal to the number of input channels on the audio device you're using. If your audio device has, say, four microphone inputs and four line level inputs, then each Mic cue will have eight possible inputs, and thus eight rows in the matrix mixer.

So, to use reverb on a Mic cue when using that device, you either need a reverb plug-in that supports eight channels of input, or you'll need to put the reverb on an output or a pair of outputs. You can put AudioUnits on either cue outputs or device outputs; we recommend cue outputs in this case, and you'll see why later on.

For this How-To, let's suppose you're using an audio interface with a microphone plugged into input #1. Let's also suppose that you're using only two speakers, plugged into the interface's output #1 (left speaker) and output #2 (right speaker.)

We're going to use Apple's *AUMatrixReverb* which requires two channels of input.

1. In QLab, open Settings and choose Mic from the list on the left.
2. Click the *Edit Patch* button next to the Mic Patch that you're using. This opens the Patch Editor for that Mic Patch.
3. In the *Device Routing* tab, note that cue outputs 1 and 2 are routed to device outputs 1 and 2.
4. Route cue output 3 to device output 1 and cue output 4 to device output 2. The initial result of this means that sending audio to cue output 1 or 3 will send that audio to the left speaker, and sending audio to cue output 2 or 4 will send that audio to the right speaker. Remove any other routing.
5. In the *Cue Outputs* tab, in the row for cue output 3, click on the drop-down menu that says *1 channel* and change it o *2 channels*. This links cue outputs 3 and 4 together as a stereo pair, which will accommodate AUMatrixReverb's requirement.
6. Still in the row for cue output 3, click on the drop-down menu that says *Add effect...* and select *Apple > AUMatrixReverb*.
7. Click the *Edit* button for the effect, and adjust the effect to taste. Important: keep the *mix* parameter at 100%, which is to say all reverb, no dry signal.
8. Close the effect editor, and click *Done* in the bottom right corner of the Patch Editor.
9. Click *Done* in the Settings view of the workspace to flip it back around and see your cue list.
10. In the cue list, create a new Mic cue.
11. With that cue selected, look at the *Device & Levels* tab of the inspector. You should see that the slider handles for cue outputs 1 - 4 are all colored yellow, meaning that they're all routed to device outputs.
12. Now, when you route audio from input 1 (row 1 in the matrix mixer) to outputs 1 and 2, you'll get dry signal from your microphone. When you route

to outputs 3 and 4, you'll get reverberant output. You can therefore think of outputs 3 and 4 as an effects bus, or auxiliary send, and vary the amount of reverb by varying the mixture between outputs 1 and 2, and outputs 3 and 4. This is why, way back at the beginning of this How-To, we suggested using reverb on cue outputs, rather than device outputs.

13. Finito!

Fade Audio Groups

When a Fade cue targets a Group cue, QLab attempts to apply the result of that fade to all the child cues within the Group. Because the starting state of each child cue may be different, Fades which target Groups can only be relative fades.

This is a problem when your goal is to fade in several sound cues from `-INF` to some audible level, because `-INF` plus any amount is still `-INF`. What to do? The trick lies in starting your cues just slightly louder than `-INF`.

1. Create several Audio cues and place them within a Group cue to your taste.
2. Set the levels of all the Audio cues in that Group such that their master levels are the same. For now, let's agree on `-10` for that level.
3. If `-10` is too loud or too soft, use the individual cue output levels for each Audio cue to get the right loudness.
4. Once all the Audio cues sound good, bring each of their master levels to 1 db louder than the minimum volume level established in Audio settings. By default this is `-60` so if that has not been changed, set your Audio cues to `-59`.
5. Create a Fade cue targeting the Group cue, and in the Levels tab of the inspector, set the master level of the Fade cue to `+49`.
6. Now, when that Fade cue is triggered, the Audio cues within the Group will have 49 dB added to their level, bringing them to `-10` as planned.
7. Do be very careful not to accidentally run that Fade cue twice while the Audio cues are playing. Since relative fades are additive, and the Audio cues are now playing at `-10`, you'll end up with very very loud sounds.
8. Die Einde!

Fade Preshow Music

Lots of folks like to use a Group cue full of Audio cues set to auto-follow for preshow music, and then use a single Fade cue targeting the Group to fade out and stop the preshow music when necessary. This works perfectly well. Another common need with preshow, however, is not quite so simple, and that's the case of lowering the level of preshow for an announcement, and then fading it back up. Everything is great until that one night when you fade down during one song, and then the song ends and the next song begins before the fade back up. When this happens, the following song starts off playing at its original volume, which is louder, non announcement level. What to do?

The secret lies in this fact: Fade cues can only fade levels on a target cue which is either running, paused, or loaded. So...

1. Create your Group cue for preshow music, with all the cues set to auto-follow, and all levels set as you like them at the "loud" level.
2. In the Basics tab for the Group cue and for each Audio cue, make sure the **Auto-load** check box is checked.
3. Now, when the Group cue is started, all the Audio cues within it will load, and thus be subject to any Fade cues which target the Group.
4. Färdig!

Fade Video Groups

When a Fade cue targets a Group cue, QLab attempts to apply the result of that fade to all the child cues within the Group. Because the starting state of each child cue may be different, Fades which target Groups can only be relative fades.

This is a problem when your goal is to fade in several Video cues from 0% opacity to, say, 100%, because 100% of 0% is 0%. What to do? The trick lies in starting your cues at 1%.

1. Create several Video cues and place them within a Group cue to your taste.
2. Set the opacity of each Video cue to 1%.
3. Create a Fade cue targeting the Group cue, and in the Geometry tab of the inspector, set the opacity to 10,000%.
4. Since 10,000% of 1% equals 100%, running the Fade cue will bring all the Video cues in the Group to 100% opacity.
5. Finis!

Replace a Projector

When you replace a screen, such as when you move from using a monitor in rehearsal to using a projector in the theater, or when you have a mechanical problem and need to trade out one projector for another, QLab does not necessarily know what's going on, and your Video cues can all appear broken with the warning "there is a problem with the cue's surface."

Here's how to tell QLab to use the new screen in place of the old one.

1. Connect your new projector to the Mac, turn it on, and make sure that it is communicating with the Mac properly.
2. Open your workspace, go to Settings, and choose Video from the list on the left.
3. Click the *Edit* button next to the name of the surface that your cues are assigned to use.
4. In the Surface Editor, select the screen that you want to reassign from the list on the left side.
5. Click *Replace screen* at the bottom of the list, and choose the new projector.
6. If the projector is a different resolution than the previous screen, you might want to change the dimensions of the surface to match the new projector, which you can do in the upper right corner of the Surface Editor.
7. Owarimashita!

Move Workspaces

1. With your workspace open, choose *Bundle Workspace…* from the **File** menu. This will start the process of making a copy of your workspace and all targeted media files to a new folder. This is not some kind of special folder or anything secret, it's just a regular folder with a copy of your stuff, which we refer to as a bundle in this context.
2. Choose a name and a location to save the bundle. This location could be, for example, on your Desktop.
3. Once QLab is done bundling, you can copy the ensuing bundle to the other Mac, via a network, a flash drive, or a hard drive.
4. Perfecta!

Once you've copied the bundle to the new Mac, don't move the workspace file out of the folder. If you want to do that for any reason, be sure to open it and then save it at least once first, which will permit QLab to correctly identify the associated media files. After that happens, QLab is able to keep track of the relative locations of all the files. You can then move things anywhere you like.

When bundling, QLab will also generate lists of the AudioUnits used in the workspace, if there are any, and the fonts used in Titles cues.

It's important to remember that only media that's targeted by a cue in the workspace will be copied.

But here's the thing: If you keep your workspace in a folder, and you also keep *all* the media for your workspace within that folder, or in sub-folders within that folder, then *you do not need to bundle a workspace to move it*. QLab will

remember the relative locations of files within the folder that contains the workspace.

This is in contrast to how QLab 2 and earlier versions of QLab 3 worked. Trying this trick with QLab 2 or QLab 3.0.x would not have gone well. Trying it now will work.

Move Licenses

1. Connect both the “old” Mac and the new one to the Internet.
2. On the old Mac, open QLab, and choose *Manage Licenses...* from the **QLab** menu.
3. Click *Deauthorize and Delete this License*
4. Quit QLab.
5. On the new Mac, install QLab by downloading it from figure53.com/qlab/download. If you need a version of QLab other than the current version, you can find it at figure53.com/qlab/download/archive (although you should only do this if you’re quite certain you need to.)
6. Put your license file onto your new Mac. You can find it attached to the email we sent when you first purchased it. If you cannot find that email, write to support@figure53.com and we’ll help you out.
7. Double-click on the license file, and it should launch QLab and install itself.
8. When prompted, authorize your new Mac for the license.
9. You’re done!

Trigger QLab With Timecode

The trick to making QLab respond to incoming timecode is that you have to enable timecode at the cue list level first:

1. Open the cue list sidebar by choosing “Cue Lists & Active Cues” from the View menu.
2. Select the cue list that you want to listen to timecode.
3. Navigate to the Timecode tab in the inspector.
4. Check the box marked “Trigger cues in this list from incoming timecode.”
5. Set the mode, sync source, and format to match your timecode.
6. Tudo feito!

Now, cues within that cue list can respond to timecode triggers.

Important: QLab can be triggered by timecode, but does not *chase* or *lock* to timecode. Pausing incoming timecode will not pause QLab, and rewinding or fast-forwarding timecode will not scrub cues. QLab just listens to the timecode as it comes in, and if a frame comes along that a cue has set as a timecode trigger, then QLab triggers that cue.