How to Use the Combination of QLab and miniStageConsole on Mac OS X

QLab is a Show Control application for Mac OS X. It allows you to control multimedia including: audio, video and MIDI from a single cue list. QLab is commonly used for playing theater sound effects cues. Since QLab (with the Pro MIDI license) can transmit MIDI commands as part of a cue, it can also control other devices and programs.

miniStageConsole is a free Stage Lighting control application. It allows you to create cues and control dimmers and lights using the industry standard DMX lighting control protocol. miniStageConsole can receive MIDI commands to tell it to execute cues, control on-screen channel faders and buttons as well as simulate keyboard commands.

Since QLab and miniStageConsole both speak MIDI, they can be connected to create a single system within one Mac computer to control multimedia and lighting.

Mac OS X supports MIDI in a way that allows MIDI from one program to be sent to another program without using a separate MIDI interface device.

Setup

1. Download and install on your Mac: QLab (from www.figure53.com) and miniStageConsole (from www.chromakinetics.com/DMX)

2. A useful free tool for figuring out what's going on with MIDI on a Mac is: snoize MIDI Monitor (free download from www.snoize.com)

3. Startup miniStageConsole.

4. On the Edit menu. choose Config MIDI...

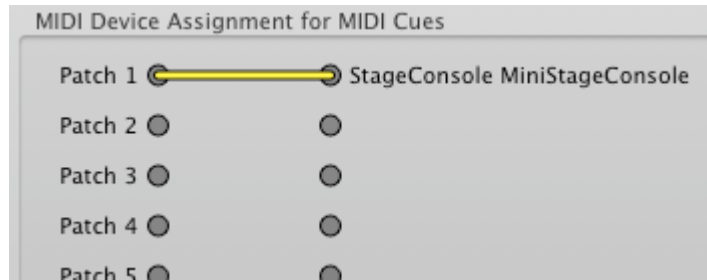5. check the Enable MIDI box. Input and Output do not matter. Click OK.
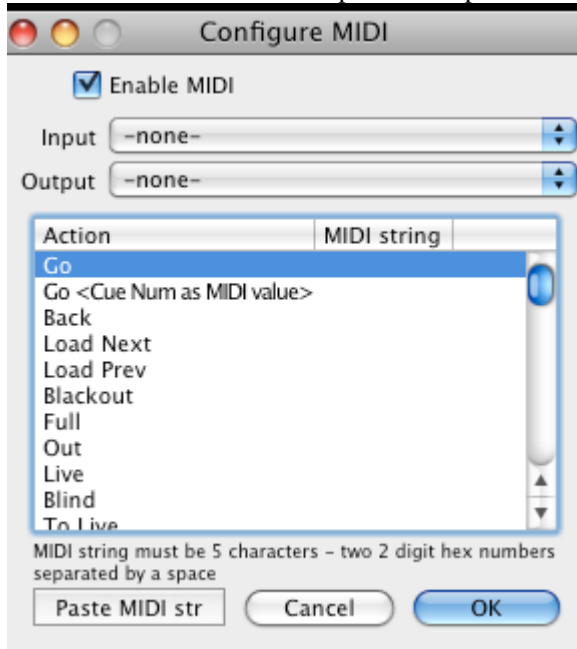
6. Now startup QLab. (Leave miniStageConsole running)

7. Click on the Preferences icon (or in The QLab menu, choose Application Preferences...)

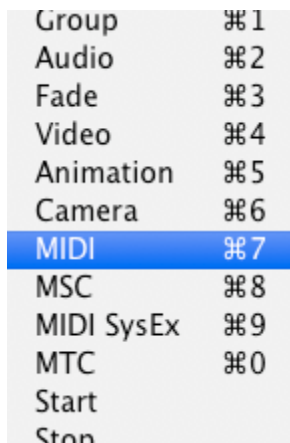
    Preferences

8. Choose MIDI from the list on the left.

9. Click on the dot next to "Patch 1" and drag a line to connect to the dot next to "StageConsole".
    Click the Done button.

MIDI Device Assignment for MIDI Cues

Patch 1 ●————————● StageConsole MiniStageConsole
Patch 2 ○   ○
Patch 3 ○   ○
Patch 4 ○   ○
Patch 5 ○   ○

10. This sets up QLab to be able to transmit MIDI to the miniStageConsole program. <image> If you do not see "Stage Console" listed, see Troubleshooting below

Creating Lighting Cues in QLab

1. on the Cue menu, choose MIDI. This will create a new MIDI cue (number 1).

|	
Group	⌘1
Audio	⌘2
Fade	⌘3
Video	⌘4
Animation	⌘5
Camera	⌘6
MIDI	⌘7
MSC	⌘8
MIDI SysEx	⌘9
MTC	⌘0
Start	
Stop	

2. Click on the MIDI message tab to set the message details.
   You may give it a name like "Light Cue 1"

3. Set the following:

- Output Patch: 1 StageConsole
- Command: Note On
- Channel: 1 (does not matter)
- Note Number: 1 (you may choose any note number)
- Velocity: 127 (click on the little down arrow to quickly go from zero to 127)

4. Arrange your screen so that you can see both miniStageConsole and QLab with QLab on the right. make sure cue 1 is highlighted (click on it) in QLab. The black box to the right of the large GO button should show "1: Light Cue 1".
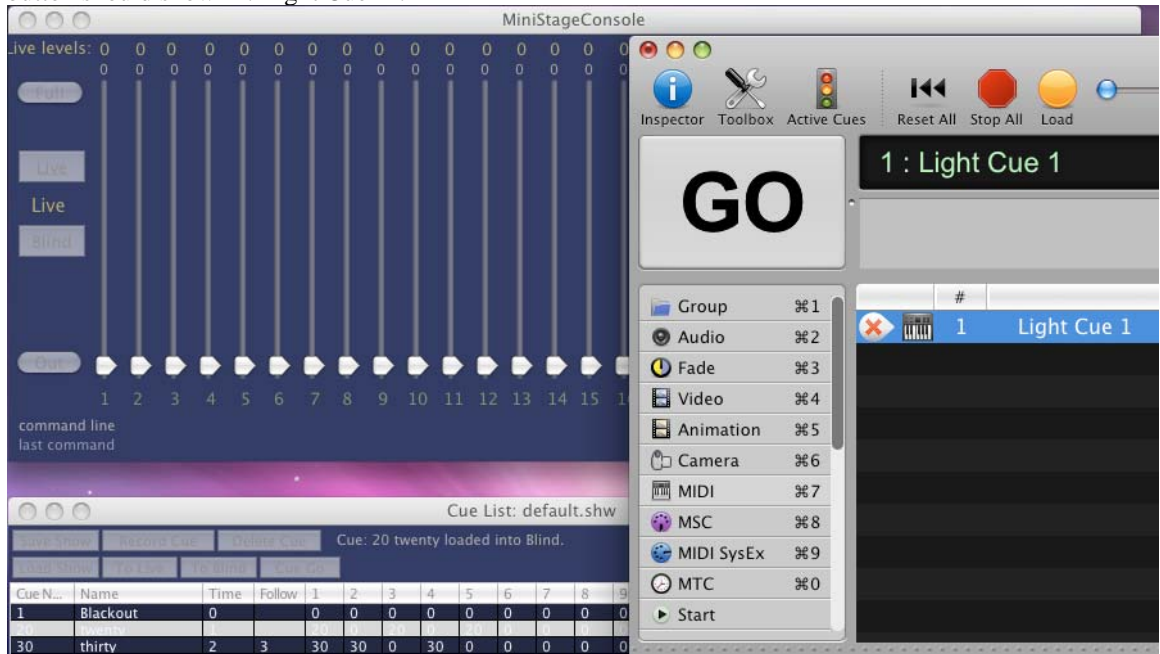
5. Watch the lower left corner of the miniStageConsole fader window (where it says "Last Command") and click the large Go button on QLab. You should see "MIDI: 90 01 7F" appear in miniStageConsole's lower left corner. This is the MIDI Note On message received from QLab.

You create a separate MIDI cue in QLab for each lighting cue you have in your miniStageConsole show cuelist.
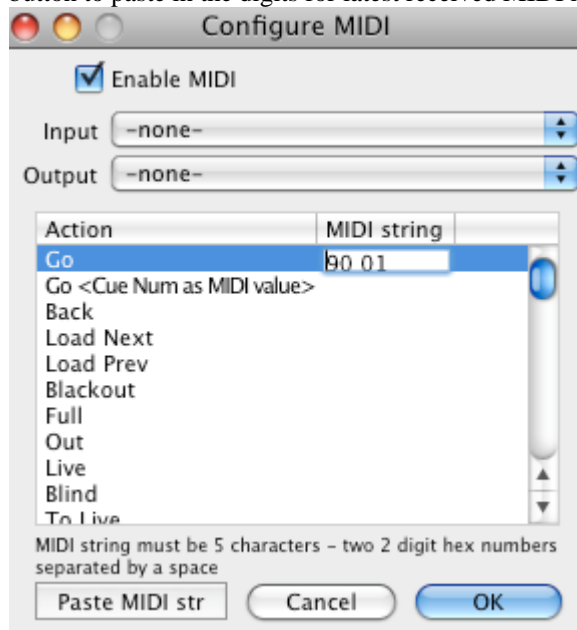
Configuring MIDI in miniStageConsole

Now we must tell miniStageConsole what to do when it receives MIDI messages.

We are going to assign the MIDI Note On message for Note Number 1 to the Go action in miniStageConsole. This means that the MIDI message we setup in QLab will cause the Go button to be activated in miniStageConsole.

1. On the Edit menu. choose Config MIDI...

2. Select the Go action in the list. type 90 01 in the MIDI string column (or click the Paste MIDI str button to paste in the digits for latest received MIDI message)

3. Click the OK button.

The Config MIDI dialog allows you to assign MIDI messages to lots of other actions, but Go is the most important one for use with QLab.

Trying it out

Assuming you have some cues in your show in miniStageConsole, we can now have QLab run them. Simply select the MIDI lighting cue you want to run in QLab and press the large GO button. You will see miniStageConsole start its cue.

Controlling specific lighting cues

QLab can tell miniStageConsole to run a specific number cue. There are two ways to do this.

1. QLab sends a Note On MIDI message and the velocity number determines the cue number in miniStageConsole. This only works if your miniStageConsole cues have no decimal points and range from 1 to 127. To use this, you assign the Note On message to the Go <Cue num as MIDI value> action in the miniStageConsole Config MIDI dialog.
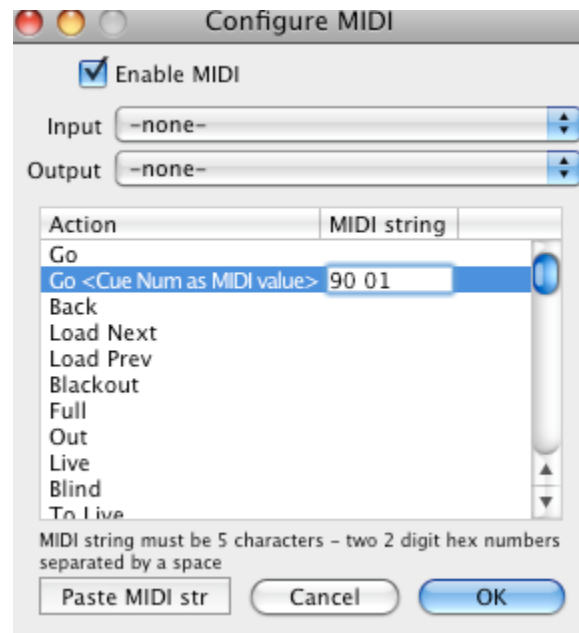
2. The second option is to transmit multiple MIDI messages from QLab to simulate miniStageConsole keyboard commands. Each digit, the decimal point and the G key can be assigned a different MIDI message as an action in the Config MIDI dialog. To tell miniStageConsole to run cue 100, send the 4 MIDI messages assigned to 1, 0, 0 and then G key in that order.
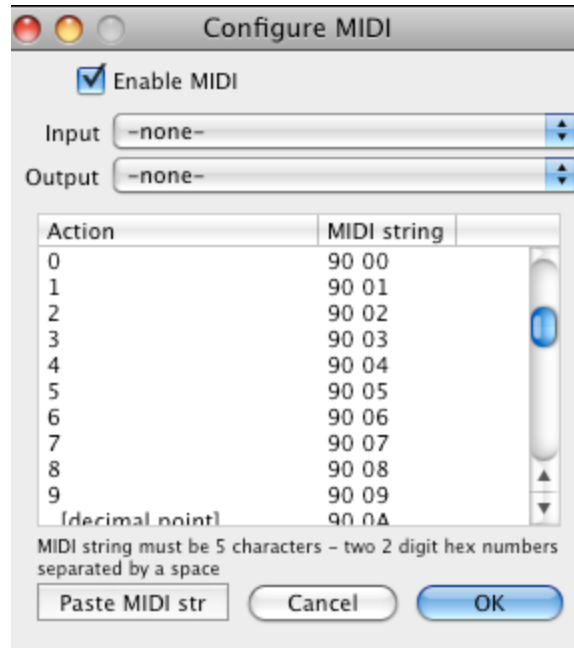

Decimal and Hexadecimal numbers

QLab uses decimal numbers when specifying MIDI notes and velocities. miniStageConsole displays these numbers as two digit hexadecimal (base 16). This means note number 10 (decimal) would be 0A in hexadecimal and note number 16 would be 10 in hex. You can always watch the MIDI message display in the lower left corner of the miniStageConsole fader window to see what is being received in hex.

Troubleshooting

QLab must always be started after miniStageConsole is already running; otherwise it will not see miniStageConsole as a MIDI Output port.

Use MIDI monitor to inspect the MIDI setup. (Make sure miniStageConsole is running first). Click on Sources and "Spy on output to destinations". You should see "miniStageConsole" listed as a destination. If not, something is wrong and miniStageConsole could not tell OS X that it is prepared to receive MIDI. Try restarting miniStageConsole and then restart MIDI monitor.
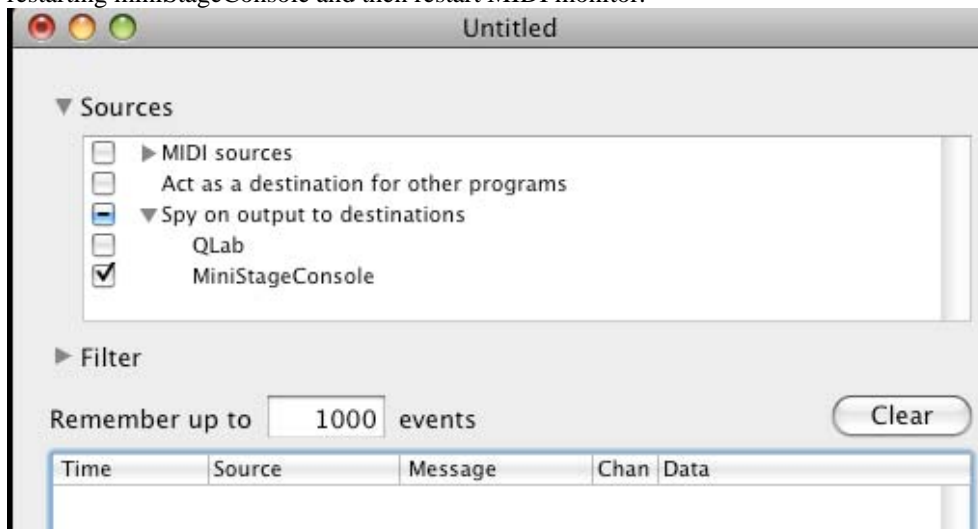
Check the check box next to miniStageConsole in the "Spy on output to destination" list. Now when QLab transmits to miniStageConsole, MIDI monitor will display the message as it is sent. miniStageConsole should now be receiving the message as well.