

Cycling '74 has transferred some of its older website content, including this page, to PDF format.

As a result, some of the information may be out of date. If you own the product below and are experiencing difficulties that this page doesn't cover, please contact our Support team (<https://cycling74.com/contact-support/>).

Helpful links:

- <https://cycling74.com/support/challenge/>
- <https://cycling74.com/support/ilok/>
- <https://cycling74.com/support/faq/>

Product: FAQ: Jitter

- [General Questions](#)
- [Technical Questions](#)
- [Cameras and Live Video Input](#)

General Questions

How do I authorize?

Jitter is part of Max. To authorize Jitter version 1.6 or earlier, [visit the Max 4 authorization page](#) for detailed instructions. To authorize Jitter version 2 or later, [visit the Max 5 authorization page](#) for detailed instructions.

What operating systems does Jitter support?

Jitter runs on Macintosh OS 9 and OS X, and Windows XP and Vista.

What version of Max/MSP is required to use Jitter?

Jitter for OS 9 requires Max/MSP 4.0.9 or higher (version 4.1 is recommended). Jitter for OS X requires Max/MSP 4.2 or higher. Jitter for Windows XP requires Max/MSP 4.3.1 or higher.

As of Max 5, the latest version, Jitter 1.7, is installed by default and no further installation is necessary.

Where can I find the Jitter 1.7 documentation?

Jitter 1.7 is part of the Max 5 installer, and the [documentation is available online](#).

Will Jitter affect the use of other video software for Max/MSP?

No. Jitter may be used side by side with other video software for Max/MSP. In fact, there are third-party bridge objects available that will let you use Jitter with the software you already have.

How does Jitter compare to other video software for Max?

Jitter is unique among other video software packages available for Max in that it is not video-specific. Jitter is designed for general data processing and, although it is optimized for real-time video, Jitter can represent (and process) OpenGL geometry, GPU-based graphics, audio, physical models, state maps, generative systems, text, or any other type of data that can be described in a matrix of values.

All of the packages available have varying profiles. For instance, softVNS 2, Cyclops and Eyes specialize in different types of image analysis. Cyclops is an all-in-one object, while the other packages are modular.

Technical Questions

How do I make Jitter output to a 2nd (or 3rd or 4th) screen?

If your video card supports extra displays, or if you have multiple video cards, you can use those extra outputs for Jitter display. The simplest way to do it is to create a jit.window object and drag the window to the display you want to use. Sending the message 'fullscreen 1' to the jit.window object will cause the window to resize to fill the entire screen of that display.

If you want to get fancy, you can use the 'jit.displays' object to determine how many displays you have attached, and where they are. Then, you can auto-position windows and fullscreen them without having to drag windows around. The 'autodetect_monitor2.pat' example patch that ships with Jitter demonstrates one such method.

My computer won't support a 2nd display, but it has a FireWire port. Can I use jit.qt.videoout to send video data to a DV camera, and then display it on a 2nd screen?

On Macintosh you can, but it will be a lot slower than using a real 2nd display because the data has to be DV-compressed before it can stream over the port. Even with a fast computer, this is a slow process.

On Windows, jit.qt.videoout is not currently supported.

Do you have any tips for maximizing Jitter's framerate?

We can't possibly answer this question completely in this space. There are many ways of optimizing your Jitter patches for greater performance. Here are two to get you started:

- If your patches use little jit.pwindow objects as monitors, and the data being displayed on them is larger than the dimensions of the jit.pwindow objects (e.g. you're monitoring a 320x240 matrix on an 80x60 jit.pwindow object), turn OFF 'use onscreen' in the object's inspector. (Why? Offscreen mode uses custom routines which are faster, in the case of downsampling).
- If your patches use Jitter objects with multiple inputs (such as jit.chromakey), you can eliminate a copy step by using the @in2_name (or @in3_name, etc.) attribute to force the object to use the specified matrix in-place (don't forget to disconnect the patchcord from the right inlet of the object). Look at the 'jit.chromakey-pile.pat' example for a demonstration of this technique.

Is there any way to get the audio from a DV device into MSP using jit.qt.grab?

No, there are technical reasons why this isn't possible right now. You can record DV audio to disk using

jit.qt.grab, but if you really need to get the audio into MSP, you'll want to route the audio output from your camera or deck to an actual audio input and use an MSP adc~ object to acquire it.

How do I record OpenGL scenes to disk as a QuickTime movie using jit.qt.record?

You need to generate a 2D raster of the OpenGL data. Jitter offers a few ways to do this, and one of them will probably work for your application.

- Instead of rendering to a jit.window object, you can render to a jit.matrix object (it must be a 4 plane char matrix, named identically to the jit.gl.render object you're using). You can then record the output of the jit.matrix object using jit.qt.record. At this time, rendering to jit.matrix is done in software, so it's quite a bit slower than rendering to a jit.window.
- Render to a jit.window, and use jit.desktop to capture the portion of the screen occupied by the window. You can then record the output of jit.desktop using jit.qt.record.
- Use jit.gl.sketch's 'glreadpixels' message to write the data into a jit.matrix, and record the output using jit.qt.record.

All three of these methods are demonstrated [in this patch](#).

Cameras and Live Video Input

Do you recommend specific video input hardware for use with Jitter?

While sidestepping the sticky issue of endorsement, the following guidelines might help you to make an informed decision if you're planning a purchase.

- Uncompressed FireWire devices offer a few advantages:
 - Low latency
 - Relatively high frame rates
 - High bandwidth port
 - Under OSX, one can generally use multiple instances of a single device
- PCI video cards, in general, provide very good performance for analog video input to desktop machines. Note: Formac ProTV card users should use vmode 1 in jit.qt.grab, which solves an issue we were seeing (note: the ProTV does not have drivers for OSX).
- DV Firewire devices, while common, provide merely average performance due to the overhead of DV decompression. As processors and QuickTime evolve, this is less and less of an issue, but a DV camera will always be slower than a similar uncompressed device.
- USB devices are generally inexpensive, but putting aside the bandwidth difference between USB and FireWire ports for a moment (which is significant), every USB webcam we've seen uses some form of compression. As the computer receives the image sequence from the camera, it must decompress each frame before performing further processing in Jitter. Thus, USB cameras are at an immediate disadvantage for frame rate and latency. The port bandwidth, then, adds a further slowdown. Generally, USB devices require 3rd party drivers, and these rarely support multiple instances of a particular device. All together, what you save in cash, you might lose in overall performance and flexibility.

For further advice and recommendations regarding specific devices, we recommend consulting the Jitter forum. As many members of the community have had opportunities to work with a variety of devices, they will be able to provide more current and in-depth testing information than we could ever hope to.

Can I use more than one video input at a time with jit.qt.grab?

Yes, with the caveat that most drivers do not allow multiples of the same device. On all operating systems, you can use as many unique devices as you can attach to your computer. Additionally, under OSX, you can use multiple instances of the same device with some drivers (e.g. 3 Unibrain Fire-I cameras). Apple's IIDC drivers support multiple devices, but at the time of this writing, Apple's DV drivers do not. OS9 does not support multiple instances of a single device type.

My video input device isn't working in Jitter! What gives?

In general, any video device that works with QuickTime on Mac (or Direct X on Windows) should work with Jitter. If your device isn't working, there may be a driver problem, or a hardware problem, or (although we really hope not) a problem with jit.qt/dx.grab. Here's a quick list of things to try, in the approximate order with which you should try them.

- If you're on windows, we recommend the use of jit.dx.grab as it doesn't require the use of third-party compatibility drivers.
- Reboot your computer. This solves 90% of these types of problems. If you're using a Macintosh, zap your PRAM for good measure (hold down ctrl-command-p-r just after you hear the startup chime, until the computer restarts again).
- Download the HackTV sample. Try running the application in the HackTV folder. Does your device work? If no, keep reading. If yes, please send a note to us with some information about your device, operating system, Max/Jitter version, etc.
- Deinstall (if possible) and then reinstall the newest version of your device's driver. Many of the problems we've seen with (especially USB) video inputs have been traced to old or unreliable driver software.
- If your device is still not working in Jitter, nor in other software, you might be facing a hardware failure of some sort. Try finding another device to test with. If you can't get that one to work, you might want to contact the manufacturer of your computer hardware for advice. If the other device works, but yours still doesn't, get in touch with the device's manufacture for support.
- If all else fails, feel free to contact us. Troubleshooting hardware/software conflicts can be frustrating and time-consuming. We're happy to lend a hand, but in most cases we've come across, the problem lies beyond Jitter.