

Remote Band Rehearsals

The information below is no longer being actively maintained, but may still be useful as a starting point and to guide the reader to other valuable online information resources.

Last updated October 5, 2024.

Quick Start

If you're on a computer (not a mobile device) and don't see the Table of Contents on the left, click the box icon to display it. On a mobile device, you'll just have to scroll down.

This document is long - here's some links to help you get started:

- **If you are a musician** and want to rehearse or perform live with other musicians via the internet, read [Latency & Solutions](#), [How I Use Them](#), and [Equipment](#), then the appropriate sections for [Real Time Applications](#) and [Delayed Sync Applications](#) of interest to you
- **If you are a choral singer** and want to rehearse or perform live with other singers via the internet, read [Latency & Solutions](#), [How I Use Them](#), and [Jamulus For Choral Groups](#), then you may also want to read [Equipment](#) and portions of [Jamulus](#).
- **If you want information on Jamulus**, read [Latency & Solutions](#), [How I Use Them](#), [Equipment](#), and [Jamulus](#).
- **If you want information on the Jamulus WorldJam**, including important notes for performers, read [Jamulus WorldJam](#)
- **If you want to route sound from Jamulus** (or another Real Time or Delayed Sync application) **to Zoom, Skype, Microsoft Teams, or another video conferencing application**, read [My Config - Using Microsoft Teams](#)

Background

I'm a relative beginner keyboard player, with a basic understanding of music theory, chord structure, etc. I play in a live-performance cover band ([The Real Imposters](#)) with my wife, and we're both also choral singers. During the 2020 COVID-19 pandemic, all musical groups had to discontinue in-person rehearsals, and I sought out solutions for remote band rehearsals, some of which are also applicable for remote choral rehearsals.

This document compiles resources that I've found and what I've learned about enabling remote band rehearsals via the internet. Although primarily focused on groups of 6 or fewer people, the information can be applied to larger groups, such as school bands or orchestras, choruses and

choirs, etc. - I've seen groups of 30+ musicians/singers performing "live" together remotely. Any additions/corrections can be sent to [Chris Rimple](mailto:chrisrimple@yahoo.com) at chrisrimple@yahoo.com.

Latency & Solutions

"Latency" is the delay from when one person makes a sound (such as playing an instrument or singing) until another person hears it. Latency is measured in milliseconds (ms), which is thousandths of a second. The human brain can perceive a delay of 15-20ms or more, and it can become difficult for musicians to stay in sync when latency exceeds 50ms. 1ms of latency is roughly equal to 1 foot of distance (for sound traveling through the air), so 25ms of latency is the same as trying to play an instrument in sync with someone that is 25 feet away. Because the speed of light is much faster than the speed of sound, large orchestras, choral groups, and marching bands (all with performers spread over a large performance area) use a "visual metronome" to stay in time - often this is a director/conductor. Smaller groups, like a band on stage, convert their sound to digital format, which can be transmitted over cables (at slightly less than the speed of light, but much faster than the speed of sound) to monitors/amplifiers for the other band members to hear.

When two people are separated by hundreds or thousands of miles and the sound is transmitted via the internet, the latency can become quite large. As a result, latency can make it very difficult for two people using a standard video conferencing application (Zoom, Skype, Microsoft Teams, etc.) to play or sing together - they'll never be able to stay in sync. Applications designed to enable remote band rehearsals overcome the problem of latency in two ways:

1. [Real Time applications](#) solve latency by decreasing it as much as possible, usually through a combination of short physical distance between participants, high speed internet connections, and reducing/removing elements that can increase latency. These applications are potentially good for playing through an entire song end-to-end and as a replacement for in-person band/chorus rehearsals.
2. [Delayed Sync applications](#) (sometimes called "Interval" applications) solve latency by increasing it, from hundredths of a second to multiple measures. These applications are great for unstructured jamming, "noodling" on an instrument, and learning to improvise, but not good for playing through an entire song end-to-end or as a replacement for in-person band/chorus rehearsals.

Swingology band has [a short blog post with basic information on a number of the Real Time and Delayed Sync applications](#).

Wikipedia has [a page comparing the technical features of Real Time and Delayed Sync applications](#).

How I Use Them

I'm using [Real Time applications](#) for remote band rehearsals, specifically [Jamulus](#), which has proven to be the most successful Real Time application and has an excellent support community. [Jamulus](#) is becoming a popular choice for musicians and my band has been using it successfully for remote rehearsals - I've also performed on the [Jamulus WorldJam](#). When connected to servers near my home, latency is less than 40ms. Choral groups are also turning to [Jamulus](#), with a number of servers for singers now available, and even large choruses using it for rehearsals.

I've also spent some hours using [Delayed Sync applications](#), primarily [Jamtaba](#). [I've created a bunch of loops](#) to play over (as backing tracks), and I've done some jamming and improvising with musicians around the world. I've had the opportunity to hear and join great musicians playing R&B, Jazz, Rock, and a variety of electronic genres. When I can't find a server that interests me, I join an empty server, start up a loop, begin playing, and wait for others to join.

For band rehearsals, I use [my full equipment stack](#) and wired internet connection with a [Real Time application](#) - I sometimes do the same with a [Delayed Sync application](#). When I just want to jam and keep it simple, I use my Mac on WiFi, 25-key keytar with MIDI adapter, [MainStage](#) for sound generation, and [BlackHole](#) to route sound to a [Delayed Sync application](#) - it's a very portable solution that I can use from anywhere in my house, as well as other locations, since [Delayed Sync applications](#) are less susceptible to WiFi latency.

Equipment

Regardless of whether you're using a [Real Time](#) or [Delayed Sync](#) application, the equipment requirements are identical.

Computer

Very few of the remote band rehearsal solutions listed here are available as mobile applications. Most applications have versions available for Windows, Mac, and Linux. Check the individual system requirements to determine if your computer is sufficient. As a general rule, any computer with a processor speed of 2.0GHz or faster and 4GB of RAM should work.

Important Note: Disable all background applications that might impact your computer's performance. That includes disabling WiFi, Bluetooth, cloud sync (Google Drive, Microsoft OneDrive, etc.), file backup (including Time Machine for Mac and File History for Windows), and any other applications that may be running in the background. On Mac, many background applications show an icon on the right side of the Menu Bar - on Windows, they often appear as System Tray icons on the right side of the Taskbar.

My equipment

- [Diagram of my equipment layout and cabling](#)
- [Mid-2015 MacBook Pro 15-inch \(Retina\)](#), 2.2GHz, 16GB RAM, 256GB SSD
 - For live performances, transported in a backpack and placed on a [Samson LTS50 Laptop Stand](#)

Internet Connectivity

You want the fastest internet connection you can get, for both download and upload. Some applications may say that 500kbps both ways is sufficient, but I recommend nothing less than 1mb down and 1mb up - you'll be far better off with 5mb up or greater. Most home "high speed" internet is at least 10mb down and 1mb up, so you're likely good if you've got cable or fiber internet service. You can test your internet speed at [SpeedTest.net](#).

You also want the most "stable" internet connection you can get, one with very little packet loss and jitter. For most people, fiber provides a far more stable connection than cable, so consider fiber if you have that option. Note that some fiber connections are not "to the house", but are only "to the street", and you want fiber all the way to the modem in your house, or you may not see all the benefits.

All of the applications recommend a wired connection, rather than WiFi. In my own testing, I've found that Wifi can work in some circumstances, but I don't recommend it - although I'm getting 300mb+ down and 40mb up on WiFi, I still get better results when wired - WiFi is subject to much more packet loss and jitter. If your computer is normally only connected via WiFi, you'll probably need a long Ethernet cable and maybe an adapter. WiFi can work for Delayed Sync applications, since they're less dependent on keeping latency low.

You'll want to plug the Ethernet cable directly into your cable/fiber modem, unless you've disabled Network Address Translation (NAT) and are using your own router - in that case, plug the Ethernet cable into your router.

Important Note: In addition to a wired connection, it's important to disable WiFi for all other devices during a remote band rehearsal, which can help to reduce jitter. Disable WiFi at the router or disconnect as many devices as possible. If you need WiFi enabled for a second device (such as when connected to Zoom for the [Jamulus WorldJam](#)), rename your WiFi SSID so that your other devices don't automatically connect to it, then return it to the regular name when done.

My equipment

- [Diagram of my equipment layout and cabling](#)
- [50-foot CAT6 Ethernet cable](#)

- [Anker USB-Hub-with-Ethernet adapter](#) - if you don't need the extra USB ports, there's the [Anker USB-to-Ethernet adapter](#)

Analog Instruments & Microphones

Most guitars, basses, keyboards, and microphones designed for live performance are analog instruments. If it plugs into an amplifier, or is amplified using analog microphones (such as acoustic drums), it's an analog instrument. Usually, analog instruments provide output via 1/4-inch "instrument" or XLR cables.

Headsets and earbuds designed for use with computers and phones may be analog instruments. They usually provide input/output via 1/8-inch cables. If you only want to sing (no other instruments) and aren't too concerned about quality, an inexpensive "gaming" headset or medium-quality earbuds are sufficient, and no [Audio Interface](#) will be needed.

Some instruments, such as keyboards, may offer both analog and digital output. If you're using an [Audio Interface](#), you'll have to decide whether to route the analog output to the [Audio Interface](#) or the digital output directly to the computer (via USB). For simplicity, I recommend using the analog output.

My equipment

- [Diagram of my equipment layout and cabling](#)
- [Photo of my keyboards and stand](#), and [another photo](#)
- Keyboards
 - [Yamaha P-80](#) 88-key digital piano, connected via MIDI Out to an [iConnectivity Mio4](#) MIDI In 1
 - For live performances, transported without a case
 - When not in use, covered by a [Roland Medium Stretch Keyboard Dust Cover](#)
 - [Nord Electro 6D](#) 61-key keyboard, connected via MIDI Out to the [iConnectivity Mio4](#) MIDI In 2 and via 1/4-inch Line Out to the [Audio Interface](#) Input 1
 - For live performances, transported in a [Nord Electro 61 Soft Case](#)
 - When not in use, covered by a [Roland Small Stretch Keyboard Dust Cover](#)
 - [MicroKorg](#) 37-key synthesizer, connected via 1/4-inch Audio Out to the [Nord Electro 6D](#) Monitor In
 - For live performances, transported in a [Gator GK-2110 Keyboard / FX Multi-Effects Board Bag](#)
 - When not in use, covered by a [DCFY Music Keyboard Dust Cover](#), although I'm not happy with the fit and am seeking other options
 - [Alesis Vortex 2](#) 37-key keytar (controller), detailed in [Digital Instruments & Microphones](#)

- [K&M Spider Pro](#) keyboard stand with [Telescopic Boom Arm](#) and two additional [Support Arm Set A](#) - I'm not using the included [Support Arm Set B](#) because I want all my keyboards to sit higher on the stand
 - For live performances, transported in a [K&M Spider Pro Bag](#) (included with the Spider Pro)
 - I was previously using a [Ultimate Support AX-48 Apex Pro](#) ([photo](#)) with additional [TBR-130 13" Support Arms](#), [CMP-495 Super Clamp](#), and [Ultimate Support Ultima Bag](#), but switched to the Spider Pro because it's 6 inches taller, allowing me to raise my keyboards higher (I play standing up) and the legs don't get in the way of keyboard pedals
- [iConnectivity Mio4](#) MIDI Out to the [Nord Electro 6D](#) MIDI In for sound generation - Electro provides sound for itself and the Yamaha, and passes through the MicroKorg sound
 - For live performances, transported in a backpack and placed on a [Gator GFW-SHELF0909 Gator Frameworks Utility Shelf](#) that is clamped to a [Samson LTS50 Laptop Stand](#)
- [Alesis Crimson II](#) digital drum kit, connected via 1/4-inch Line Out to the [Audio Interface](#) In 3
- Two [Audix OM5](#) microphones, connected via XLR cables to the [Audio Interface](#) In 2 and 4
 - For live performances, transported in a backpack
 - When not in use, covered by Audix OM5 bags (included with the microphones)

Digital Instruments & Microphones

Digital instruments and microphones are designed to output directly to a computer or other digital receiver, usually via a USB cable. Some of these instruments may not have a speaker or any analog output, instead functioning as a “controller” for music applications - examples include keyboards sending MIDI to applications like MainStage or GarageBand, a digital drum kit that sends MIDI to a rack-mounted synthesizer, and so on. Any microphone that connects to a computer via a USB cable is a digital microphone.

Headsets designed for use with computers may be digital instruments. They usually provide input/output via a USB cable. If you only want to sing (no other instruments) and aren't too concerned about quality, an inexpensive “gaming” headset is sufficient, and no [Audio Interface](#) will be needed.

As noted in [Analog Instruments & Microphones](#), I recommend using analog output whenever possible, for simplicity. For example, my [Alesis Vortex 2](#) keytar is a digital-only controller - it generates no sound and has no speaker or analog output. But, instead of sending MIDI Out to a music application like MainStage to generate digital sound, I send MIDI Out from the keytar to my [Nord Electro 6D](#) MIDI In, and the Nord generates analog sound which then goes via 1/4-inch Line Out to my [Audio Interface](#).

If you want to use a digital instrument, you'll need to route the output from it to your [Audio Interface](#) (usually through your computer) or an application. More details are provided in [Digital Audio Workstation](#). To route the output to or from an application, you may need to use a [Virtual Audio Driver](#) for audio routing "within" your computer.

My equipment

- [Diagram of my equipment layout and cabling](#)
- [Alesis Vortex 2](#) 37-key keytar (controller), connected via WiFi dongle to the [iConnectivity Mio4](#) USB Host
 - For live performances, transported in a [Fender FEMS-610 Mini Strat Gig Bag](#)
 - When not in use, sits on an [Electric Acoustic And Bass Guitar Stand](#)
- [Rock Band 3 MadCatz 25-key keytar](#) (controller) connected via [iConnectivity Mio](#) to my Mac, when I'm traveling or keeping it simple
 - For live performances, transported in a backpack

Headphones / Speakers

You should not listen to yourself play or sing "live" - instead, you listen to yourself played back as others hear you (which is slightly delayed), helping to ensure synchronization. To achieve this, you need wired headphones, earbuds, or similar that are connected to your [Audio Interface](#) (or possibly your computer if you're doing any pre-/post-processing of sound before/after outputting to a remote band rehearsal application).

In a quiet environment, earbuds are sufficient. In a louder environment, you'll want to block out background noise with headphones. "Noise cancelling" is not necessary and not particularly helpful, unless background noise is relatively constant and high pitched (such as roadway noise). If you will be singing, you may prefer "open back" headphones that allow your own voice to reach your ears "live", or block it completely with "closed" headphones.

You should not be monitoring your "live" sound and should not be listening through speakers, particularly if you have a microphone (since the microphone will likely pick up sound from the speakers, creating a feedback loop and echo).

Important Note: Headphones/earbuds must be wired, not wireless. Bluetooth headphones/earbuds cannot be used because they add significant latency.

Important Note: For some remote band rehearsal applications, the headphones/earbuds must be capable of outputting 48KHz - some can only output lower bitrates, or have a range of output options but exclude 48KHz.

My equipment

- [Diagram of my equipment layout and cabling](#)

- I have a [Mackie SRM150 monitor speaker](#), but it's not used with remote band rehearsal applications because I don't want to hear local audio
 - For live performances, transported in a [Mackie SRM150 Travel Bag](#) and attached via a [K&M Extension Microphone Holder](#) to a [Samson LTS50 Laptop Stand](#)
- I started with [Apple EarPods with a 3.5mm connector](#), using a [3.5mm Male to Female extension cable](#) and [3.5mm to 6.35mm adapter](#), but found that I was hearing too much background noise, particularly when my wife was drumming
- I tried [Audio-Technica Earbuds](#) (in ear), using the same extension and adapter, but was still hearing too much background noise
- My wife is using [Bose QC25 headphones](#) (over ear, closed) while drumming, and they're working well for her - I tried them and found that I couldn't hear myself well enough while singing
- Most recently, I've been using [Beyerdynamic DT 770 headphones](#) (over ear, closed) and [Beyerdynamic DT 880 headphones](#) (over ear, open back) to see if they are good for keeping out drum noise while allowing me to sing comfortably. The DT 770s keep out the noise, but I couldn't hear myself well enough while singing, so I've settled on the DT 880s and have been quite happy with them.

Audio Interface

An Audio Interface ([AI](#)) converts audio signals from analog-to-digital and digital-to-analog. The [AI](#) connects to a computer and one or more analog devices (guitar, bass, keyboard, microphone, etc.) - anything played on the device is transmitted in digital form to the computer, where it can be recorded or passed on to a remote band rehearsal application. As well, anything played on the computer is transmitted to the [AI](#), where it can be output to analog amplifiers and other analog devices.

Important Note: For some remote band rehearsal applications, the [AI](#) must be capable of outputting 48KHz - some [AIs](#) can only output 44KHz, or have a range of output options but exclude 48KHz.

Example AIs

- [Behringer UCG102](#) is good if you just want to convert a single analog device to digital, such as a guitar or bass
- [Focusrite Scarlett Solo](#) and [Focusrite Scarlett 2i2](#) are good if you want to convert multiple analog devices at the same time, such as a guitar and microphone

My equipment

- [Diagram of my equipment layout and cabling](#)
- [iConnectivity AUDIO4+](#), which allows me to convert 4 analog devices at once - currently, those are keyboards (via 1/4-inch analog out), digital drum kit (via 1/4-inch analog out), and two microphones

- For live performances, transported in a backpack and placed on a [Gator GFW-SHELF0909 Gator Frameworks Utility Shelf](#) that is clamped to a [Samson LTS50 Laptop Stand](#)

Digital Audio Workstation

A Digital Audio Workstation ([DAW](#)) is an application for creating and editing digital multi-track recordings. DAWs can be used with [Digital Instruments & Microphones](#), or [Analog Instruments & Microphones](#) connected via an [AI](#). A [DAW](#) will only be necessary if you are using one or more [Digital Instruments & Microphones](#) or need pre-/post-processing of sound before/after outputting to a remote band rehearsal application.

Common DAWs include Logic Pro, MainStage, GarageBand, Audacity, Reaper, Pro Tools, Cakewalk, Cubase, and others. Note that Audacity (Windows, Mac) , Reaper (Windows, Mac), and GarageBand (Mac) are free - others may be as well.

Important Note: For some remote band rehearsal applications, the [DAW](#) must be capable of outputting 48KHz - some [DAWs](#) can only output 44KHz, or have a range of output options but exclude 48KHz. GarageBand cannot output 48KHz.

My equipment

- [Diagram of my equipment layout and cabling](#)
- [MainStage](#) when I want to generate sound for a digital instrument - other DAWs can do it, but MainStage is designed specifically for live performance so adds very little latency
- [Reaper](#) when I want to use [Ninjam](#) with multiple backing tracks
- [Audacity](#) or [Reaper](#) when I want to load Jamulus recordings saved by a Jamulus server
- [GarageBand](#) for quick creations, such as looping backing tracks
- [Logic Pro](#) for heavier editing with a greater feature set

Audio Routing / Virtual Audio Driver

A virtual audio driver may be needed with some band rehearsal applications if you want to (1) use more inputs than the application allows or (2) route audio from one application to another, such as when you want to use [Jamulus](#) or [Jamtaba](#) with a [DAW](#) but without an [AI](#). For example, if you want to use [Jamulus](#) (which is limited to 2 inputs) with an [AI](#) that has 4 devices plugged in (keys, drums, 2 mics), you'll need to route the audio from the [AI](#) (4 inputs) to a [DAW](#) (4 inputs) to a virtual audio driver (2 outputs) to [Jamulus](#) (2 inputs).

For Mac, virtual audio drivers include [Soundflower](#) (MacOS 10.10 Yosemite and earlier), [BlackHole](#) (MacOS 10.11 El Capitan and later, including Catalina), [LoopBack](#), and others. If you're a non-technical Mac user, you may find [LoopBack](#) easiest to use, although it's not free like some. For Windows, there's [ASIO4ALL](#), [JACK](#), [VoiceMeeter](#), and others. [Reaper](#) ([DAW](#)) for Windows includes an audio routing solution (ReaRoute) and is free.

For a little background on audio routing for Mac, see the “Using a DAW” section of Jim Amsden’s [blog post](#) and the [support page for BlackHole](#). For Windows, see the “Soundcard Settings” section of Brian Mountford’s [Google doc](#). For any platform, also see the “Routing The DAW To And From Jamulus” section of Brian Mountford’s [Google doc](#), which provides an excellent (and simple) overview of merging inputs and pre-/post-processing of audio before sending to, or when received from, a remote band rehearsal application.

Important Note: For some remote band rehearsal applications, the virtual audio driver must be capable of outputting 48KHz - some virtual audio drivers can only output 44KHz, or have a range of output options but exclude 48KHz.

My Equipment

- [Diagram of my equipment layout and cabling](#)
- [BlackHole](#)

Bitrate

For some remote band rehearsal applications, any [headphones/earbuds](#), [AI](#), [DAW](#), and [virtual audio driver](#) used must be capable of inputting and outputting 48KHz.

To check and set the bitrate on a Mac:

- From the Finder, open Applications, then the Utilities folder
- Run the Audio MIDI Setup application
- From the Window menu, choose Show Audio Devices (if the window is not already shown)
- Select each device in the list on the left and ensure that the Format or Sample Rate on the right is set to 48KHz, for both Input and Output
 - If you have options like 16-bit, 24-bit, and 32-bit, select 32-bit

Checking and setting the output bitrate on Windows depends on the device driver being used. For [ASIO4ALL](#):

- TBD

Access Permissions

From some remote band rehearsal applications, security “permission” may be required to access microphones, instruments, AI, etc.

To check and set access permissions on a Mac:

- From the Finder, open Apple > System Preferences
- Select Security & Privacy, then the Privacy tab

- Select Camera from the list on the left, then Check On any remote rehearsal application in the list on the right
- Select Full Disk Access from the list on the left, then Check On any remote rehearsal application in the list on the right
- Select Input Monitoring from the list on the left, then Check On any remote rehearsal application in the list on the right
- Select Microphone from the list on the left, then Check On any remote rehearsal application in the list on the right
- Select Screen Recording from the list on the left, then Check On any remote rehearsal application in the list on the right

To check and set access permissions on Windows:

- TBD

Real Time Applications

Real Time applications are potentially good for playing through an entire song end-to-end and as a replacement for in-person band rehearsals. These applications solve latency by decreasing it as much as possible, usually through a combination of short physical distance between participants, high speed internet connections, and reducing/removing elements that can increase latency.

Most Real Time applications use a hub-and-spoke model, requiring all users to connect to the same server. As each user plays, the sound they generate is transmitted to the server, then sent back to all users (including the original). In the hub-and-spoke model, it's critical that the latency between users and the server be as low as possible. Other Real Time applications use a peer-to-peer model, requiring all users to connect to all other users. As each user plays, the sound they generate is transmitted to all other users. In the peer-to-peer model, it's critical that the latency between all users be as low as possible. Of the two, the hub-and-spoke model is more forgiving for users with lower speed computers and lower speed internet connections.

Jamulus

Jamulus uses the hub-and-spoke model and can be very successful when all users are within 100 miles of the server. When looking for servers, Jamulus provides latency values, allowing the user to select the nearest/fastest server for the best experience. Jamulus has public servers, but it's also reasonably simple to set up a private server, including running in a virtual machine on Microsoft Azure, Google Cloud, or Amazon Web Services. My band has used Jamulus, on a server within 10 miles and at an overall latency of 30-40ms, and has had no meaningful issues staying in sync.

Jamulus is limited to 2 mono input channels, which can restrict users with multiple instruments and microphones. That limitation can be overcome by running multiple instances of the Jamulus application or by merging multiple inputs to a single input, which is then sent to Jamulus - this is usually accomplished with a [Virtual Audio Driver](#) or [DAW](#).

The [Jamulus WorldJam](#) community of musicians has developed around Jamulus, including the hosting of [WorldJam Nights](#), a regular, live audio/video broadcast of musicians playing from around the world.

General Information & FAQ

- Available for Windows, Mac, and Linux
- Requires 48KHz output (see notes in [AI](#) and [DAW](#))
- <http://jamulus.io>
- <https://sourceforge.net/projects/llcon/files/latest/download>
- <https://github.com/corrados/jamulus/wiki/Hardware-Setup>
- <https://github.com/corrados/jamulus/wiki/Sound-Devices>
- <https://github.com/corrados/jamulus/wiki/Getting-Started>
- <https://github.com/corrados/jamulus/blob/master/src/res/homepage/manual.md>
- <https://github.com/corrados/jamulus/wiki/Command-Line-Options>
- <https://github.com/corrados/jamulus/blob/master/ChangeLog>
- The INI file can be found
 - On a Mac at ~/.config/Jamulus
 - On Windows at C:\Users\yourusername\AppData\Roaming\Jamulus
- On a Mac running Mojave or later, Jamulus supports Tabs, which is particularly useful if you're running multiple instances of Jamulus
 - ["Always" enable Tabs for applications](#)
 - In Jamulus, under the View menu, choose Show Tab Bar

Installation

- <https://github.com/corrados/jamulus/wiki>
- Jim Amsden has [an excellent blog post on Jamulus installation](#) (for Mac, but most of the information is valid for all), including notes on using a guitar [AI](#) and using a [DAW](#), as well as setting up a Mac Jamulus Server
- Brian Mountford has [an excellent Google Doc on Jamulus installation and use](#) (for Windows, but most of the information is valid for all) for choral groups, including notes on using a [DAW](#) and setting up a Jamulus Server
- Simon Tomlison has [a good Facebook doc on Jamulus installation and use](#)
- Audio Geek 11 has [a good blog post on Jamulus installation](#), including notes on setting up public and private Jamulus Servers
- DistrictJam's [Jamulus installation instructions for Windows](#) and [Jamulus installation instructions for Mac](#)
- Max has [a detailed blog post on using Jamulus with the Reaper DAW](#)
- Peter Kirn has [a brief blog post about routing audio on Mac and Windows](#)

- Stephen O'Reilly has [a short Facebook doc on using Jamulus with a chorus](#)
- Anand Sitaram has [a detailed blog post on using Jamulus with a chorus](#), particularly for Windows
- MusicBridge offers [a complete solution with hardware and servers](#), but it is not free

Instructional Videos

I haven't watched all these yet and need to order them, highlight important contents, etc.

- <https://www.facebook.com/723337575/videos/10159716980357576/> - Simon Tomlinson demonstrates how to use Jamulus to jam with other musicians
- <https://www.facebook.com/groups/507047599870191/permalink/523309784910639/> - using Reaper and others
- https://www.facebook.com/groups/JamulusWorldJam/?post_id=523309784910639 - merging Jamulus audio with Zoom video
- <https://www.youtube.com/watch?v=IB4ZxDb9vnU>
- <https://www.youtube.com/watch?v=2Tqehlebsm4>
- <https://www.youtube.com/watch?v=rrkic-DKqCA> - Windows, using a DAW
- <https://www.youtube.com/watch?v=hRWIJggRUXc> - Windows, using a DAW and more
- <https://www.youtube.com/watch?v=4R2cjAyFDJ8> - Windows, using a digital instrument
- <https://www.youtube.com/watch?v=joZT33XZ0-o> - Mac, using processed audio with Soundflower
- https://www.youtube.com/watch?v=eGXSuxHIK_g - Mac, using a DAW
- <https://www.youtube.com/watch?v=HfRmCIJg7JE>

Usage Videos

- <https://www.youtube.com/watch?v=Pgu7BRHrbdw>
- <https://www.youtube.com/watch?v=2x-gwMmVK-s>
- <https://www.youtube.com/watch?v=pFgkB5lqcUQ>
- <https://www.youtube.com/watch?v=fxjXs1XswjU>
- <https://www.youtube.com/watch?v=jkpBRYotoXs>
- https://www.youtube.com/watch?time_continue=109&v=mrWAUXc-LmM - choral group

Server Installation

- <https://github.com/corrados/jamulus/wiki/Running-a-Server>
- <https://github.com/corrados/jamulus/wiki/Running-a-Private-Server>
- Simon Tomlinson has [a Facebook document with detailed instructions to configure a local private server](#)
- Simon Tomlinson has [a Facebook document with detailed instructions for installing Jamulus Server on Ubuntu on AWS](#) and another with [detailed instructions to upgrade that server](#)
- Andrew Ferguson has [a blog post with detailed instructions for installing Jamulus Server on Debian on Vultr](#)
- Tony Mountifield has a [Jamulus Server Explorer](#) that shows the Jamulus Server version number and other details

- [Melomax](#) offers private Jamulus Servers that can be reserved - at this time, the service is free

Server Instructional Videos

- <https://www.youtube.com/watch?v=xa9bq9MkK0E> - private server (local?)
- <https://www.youtube.com/watch?v=9anNmobd5aY> - on Google Cloud
- <https://handsomejacks.co.uk/index.php/set-up-a-jamulus-server-on-aws/> - on Amazon Web Services

Social Media & Forums

- Facebook page [WorldJam](#)
- Facebook group [Jamulus \(official group\)](#)
- Facebook group [Jamulus Choral](#)
- Facebook group [Digital Chord Ringers](#) - barbershop singers
- SourceForge forum [Jamulus](#)

Jamulus WorldJam

A number of musicians in Western and Northern Europe came together to create the Jamulus WorldJam (WJ), a weekly “jam session” that is broadcast live via Facebook and YouTube, using Jamulus for audio and [Zoom](#) for video. Although called a “jam”, WJ is a structured event with hosts, sound checks, and a predefined set list, but remains open to everyone and participants are not professionals.

As awareness and popularity of the WJ has grown, participation has expanded worldwide, but latency is still a limiter. Some European and USA musicians have attempted to play together, but latency often exceeds 100ms and makes staying in sync exceptionally difficult. WJ offers multiple servers with the results fed into the WJ - for example, a server in Seattle WA (USA) is used by musicians in nearby states and Canadian provinces to play together on the WJ, while servers in Iowa and Virginia are used by musicians near those states.

Songs can be added to an upcoming set list by anyone, then it’s up to that musician to recruit others to play the needed parts. Usually that’s accomplished by suggesting the song in the appropriate Facebook group, where other musicians will learn about it and volunteer. In general, the WJ observes a “maximum 2 songs per lead singer per week” rule, but no limit on instrument participation, and the organizers are mindful to give space to new participants and those that didn’t participate during the prior week. Rehearsing isn’t a requirement, but it’s common for musicians on a song to try to rehearse a few times before the WJ - if the participants don’t feel comfortable that a song is ready, it can always be rescheduled to a later date.

The set list is locked the day before the WJ, and includes sound check, show schedule, and other participant information. It’s published to the [Jamulus WorldJam](#) Facebook group.

When it's time for the WJ, first-time participants join a sound and video check server 1.5 hours before showtime, then join a waiting room server at least 15 minutes before showtime. The waiting rooms have a "virtual monitor", so musicians can hear the WJ while waiting to perform, and remain connected to the Zoom video conference but don't send video until it's time for their song(s). Some musicians run Zoom on the same computer with Jamulus, while others use a second device. If using a second device, it's best if the device is not using WiFi.

The WJ is simultaneously broadcast on Facebook (720p) and YouTube (1080p). Audio is not meaningfully altered from what's sent by the individual musicians - a limiter catches peaks, light compression reduces dynamic range, and a slight equalizer brings out a few frequencies. While Jamulus allows individual musicians (listening to the "virtual monitor") to control the balance that they hear, such as "more vocals and less guitar", the broadcasts to Facebook and YouTube are mixed live by a WJ volunteer sound engineer. Musicians on the same song can improve their balance by doing their sound checks together, ensuring a better relative balance before performing.

The WJ prefers to receive a separate Jamulus feed for each instrument or voice, which can be accomplished by [running multiple Jamulus instances](#) if needed. If a musician has only 2 inputs, such as voice and guitar, it's OK to run a single Jamulus instance with inputs split L/R, Mono output, and Pan centered.

Important Notes if you want to participate in the WJ...

- Set List settings
 - Your [WorldJam Set List](#) Jamulus Name is displayed in the credits at the beginning of each song during the WJ, but your instrument is not shown
 - Set it by logging into the [WorldJam Set List](#), then choosing My Profile from the Account menu
 - It is common for participants to set their [WorldJam Set List](#) Jamulus Name to their real name, but that is not a requirement
- Jamulus settings
 - You must be running the [latest version of Jamulus](#)
 - Your Jamulus Profile Name is not used during the WJ, but it helps the WJ staff if it is the same as your [WorldJam Set List](#) Jamulus Name
 - In Jamulus, choose My Profile from the View menu
 - Your Jamulus instrument setting helps the WJ staff during your sound check and performance
 - In Jamulus, choose My Profile from the View menu
 - To ensure that you hear all participants, **in Jamulus select Clear All Stored Solo Settings from the Edit menu**

- To connect to the WJ servers, in Jamulus select Settings and set the Custom Central Server Address to “worldjam.vip”, then select Custom from the List dropdown when Connecting
- To make it easier to switch between servers, in Jamulus select Connection Setup from the View menu and leave that window open
- Zoom settings
 - When connecting to the WJ Zoom meeting, join video **but do not join computer audio** if prompted, and disconnect from computer audio if you have joined
 - Visit the [WorldJam website](#) for Meeting ID and Passcode details
 - Your Zoom Name is displayed with your video during the WJ
 - View the Participant list, click on your name and choose Rename, then set your name as you wish
 - It is common for participants to set their Zoom name to “Name (2 Letter Country Abbreviation)”, such as “Chris (US)” - this allows viewers to see the locations of participants
 - If you are using a virtual background in Zoom, avoid dark clothing if in a dark room or light clothing in a well-lit room, and avoid clothing that closely matches the background color - this can cause Zoom to have difficulty differentiating between you and your background, causing you to “disappear” as you move around
 - After sound check, remain connected to the WJ Zoom meeting but stop Sending Video, and **do not Send Video again until instructed to do so by the WJ hosts**
- Flow of the show
 - After sound check, you will usually have some time to run through songs in a Waiting Room or Studio before the show starts, or get food, change into a costume, etc.
 - While in a waiting room or other pre-performance server, you will hear the WJ staff giving you instructions and updates, such as “performers on song X should move to Backstage” and so on
 - While in the Backstage room, you will not be muted, and should applaud when a song completes - your applause is heard by the performers and on the video stream
 - The WJ hosts will talk with you before performing, between songs, and after performing, so **don’t rush to stop Sending Video in Zoom after you’re done performing**
 - When done performing, in Jamulus select Settings and clear the Custom Central Server Address, and remember to select All Genres from the List dropdown when Connecting to non-WJ servers

General Information & FAQ

- Facebook file [WorldJam Instructions](#)
- [WorldJam Set List](#) - view the upcoming set lists, add a song, and join a song

Social Media & Forums

- Facebook group [Jamulus WorldJam](#) for all participants, but primarily those in Europe
 - [Jamulus WorldJam - Pacific NW](#) for participants using the Seattle server
 - [Jamulus WorldJam - Midwestern USA](#) for participants using the Iowa server
 - [Jamulus WorldJam - East Coast USA](#) for participants using the Virginia server
- YouTube channel [WorldJam](#)

Jamulus For Choral Groups

A number of choral groups are using Jamulus for rehearsals and “drop-in singing”, most notably [barbershop](#) choruses and quartets, which have begun hosting Jamulus servers with “barbershop” in the name. The organizers of the [Jamulus WorldJam](#) also hosted the first [Jamulus Choral Festival](#), which featured up to 19 singers at one time. By default, Jamulus servers are limited to 10 users, but some have been configured for 25 or 50 - experiments have shown that Jamulus struggles with more than 20 users, so it’s not yet a solution for larger choruses. But some tests have been done to “chain” multiple Jamulus servers, possibly allowing as many as 180 users at one time.

If you’re a singer wanting to use Jamulus, much of the [Equipment](#) section can be ignored. At a minimum, you will need:

- A [computer](#)
- A [wired internet connection](#)
- A [headset or earbuds with microphone](#), connected via the computer’s headphone jack or USB - wireless (Bluetooth) headsets and earbuds cannot be used, because they introduce too much latency

Configure your computer’s Sound settings to use the headset/earbuds for Input (Microphone) and Output (Speakers), then run Jamulus and select “in:System/out:System” for your Device. Connect to a server and speak, and you should hear your voice returned from the server. If other users are on the server, you’ll hear them and they should be able to hear you - if so, you’re ready to sing.

If you decide that you want a higher-quality microphone, there are a number of reasonable-quality “gaming” headsets available, as well as the [Blue Yeti](#) line of USB microphones. Check with other Jamulus singers to see what they’re using.

You may also want to review these instructional links:

- Stephen O’Reilly has [a short Facebook doc on using Jamulus with a chorus](#)
- Anand Sitaram has [a detailed blog post on using Jamulus with a chorus](#), particularly for Windows

Important Note: When connecting to Jamulus servers, the List dropdown at the top of the server list defaults to “All Genres”, but most Jamulus servers being used by choral groups will be found under “Genre Classical/Folk/Choral”, so try other choices if you don’t see the server you’re seeking.

Social Media & Forums

- Facebook group [Jamulus Choral Community](#)
- Facebook group [Digital Chord Ringers](#) for barbershop singers

My Usage

How I’m using Jamulus.

- [Diagram of my equipment layout and cabling](#)
- I use Jamulus for remote band rehearsals with my [full equipment stack](#) including keyboards, digital drum kit, and 2 microphones, and to participate in the [Jamulus WorldJam](#)
- I have also used Jamulus to [perform for co-workers in a Microsoft Teams web conference](#), using a [virtual audio driver](#)
- I’m running a [public Jamulus server on Vultr](#) that my band uses for rehearsals
- I’m outputting analog sound from my keyboards and the drum kit to my [AI](#), not sending MIDI to a [DAW](#) to generate digital sound
- My [AI](#) has 4 active inputs: IN1 = keyboards, IN2 = keyboardist’s mic, IN3 = drums, IN4 = drummer’s mic
- My [AI](#) has 4 outputs: OUT1 = monitor speaker, OUT2 = keyboardist’s headphones, OUT3 = monitor speaker, OUT4 = drummer’s headphones
 - I’m not using my [AI’s](#) (single) headphone output because I want separate outputs for keyboardist’s headphones and drummer’s headphones, and didn’t have a Y splitter with which I could pan signals L/R
 - My monitor speaker is turned down, so sound is only heard via headphones
- Because Jamulus allows only 2 inputs and I have 4, I run 2 Jamulus instances
 - With 2 Jamulus instances, I have separate mixers for keyboardist’s headphones and drummer’s headphones, allowing each to control volume levels heard from other musicians
 - Instead of 2 Jamulus instances, I could combine all my inputs using a [virtual audio driver](#) or [DAW](#), but wouldn’t have separate mixers

My Config - 2 Jamulus Instances

How I have Jamulus configured for my [full equipment stack](#) when using 2 Jamulus instances.

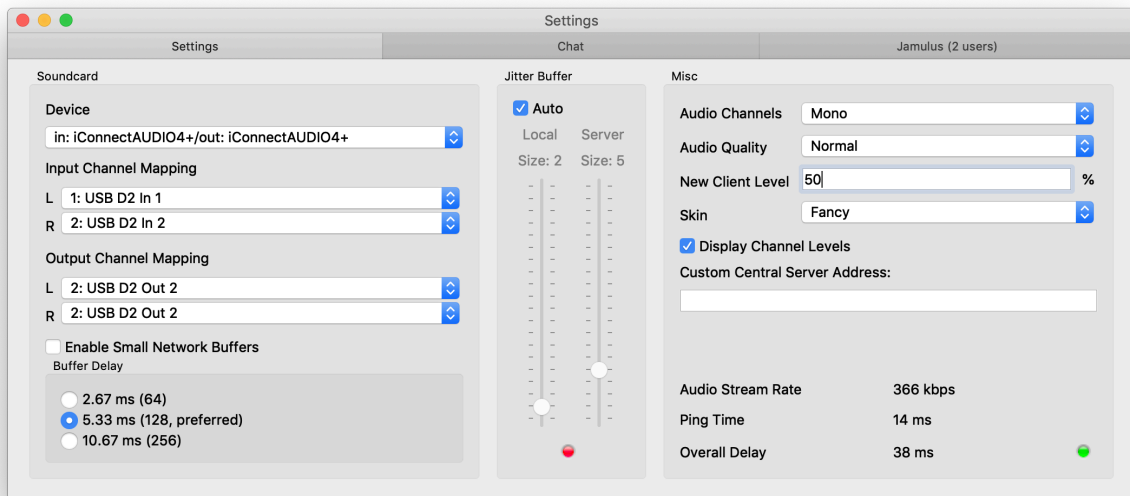
- [Diagram of my equipment layout and cabling](#)
- One-Time Setup
 - Finder > Applications > Utilities > Audio MIDI Setup > Show Audio Devices
 - Ensure that every listed device (particularly BlackHole) is set to 48KHz

- Finder > Applications
 - Right-click Jamulus, then Duplicate
 - Rename Jamulus as JamulusKeys and the copy as JamulusDrums
- iConfig > Audio Mixer
 - Analog Inputs
 - All input gains should be set to peak at 1-2 yellow LEDs in Jamulus, not red (which indicates overmodulation and clipping)
 - Input 1 volume set +20db (keyboards)
 - Input 2 volume set +27db (keyboardist's mic)
 - Input 3 volume set +27db (drums)
 - Input 4 volume set +27db (drummer's mic)
 - Sub Mix/Bussing > Analog Line Outs 1-2 (OUT2 = keyboardist's headphones)
 - All inputs split (not merged into stereo channels)
 - Sources A1, A2, A3, and A4 set to muted (no monitoring of inputs)
 - Sources U3 and U4 set to muted (drums and drummer's mic)
 - Need to check if U1, U2 should also be muted
 - Sub Mix/Bussing > Analog Inputs 3-4 (OUT4 = drummer's headphones)
 - All inputs split (not merged into stereo channels)
 - Sources A1, A2, A3, and A4 set to muted (no monitoring of inputs)
 - Sources U1 and U2 set to muted (keyboards and keyboardist's mic)
 - Need to check if U3, U4 should also be muted
 - iConfig > File > Save Preset > "online"

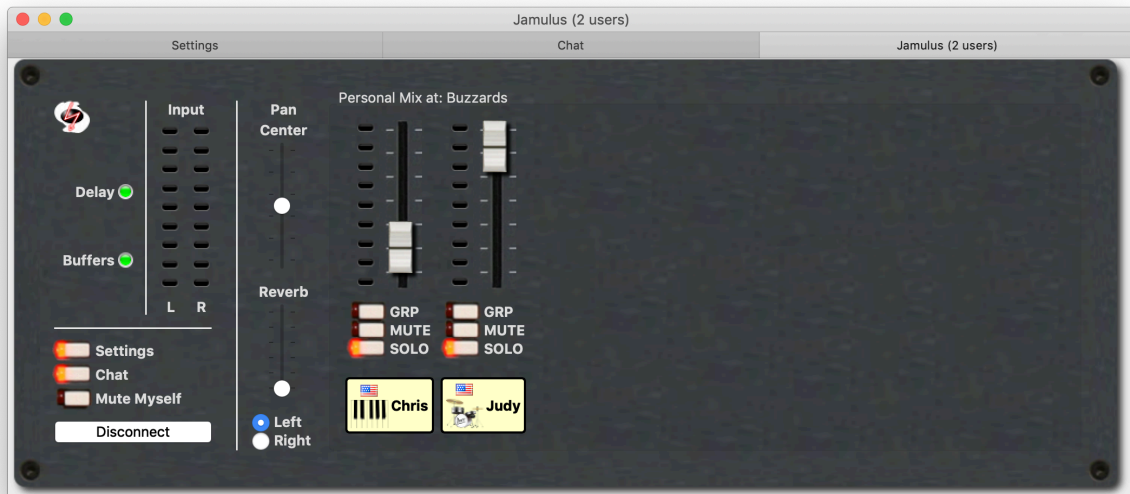


- Every Session
 - MacOS > System Preferences > Sound

- Output set to system default (usually Internal Speakers)
 - Output volume set to Mute
 - Input set to system default (usually Internal Microphone)
 - Quit System Preferences
- iConfig > File > Open Preset > “online”
 - Quit iConfig
- [Not quite working yet] Run Jamulus from a Terminal command line, so that I can specify parameters for each instance
 - Terminal remains open/busy with first instance, so can't run second
 - May be able to solve using “open -a JamulusKeys --args”, but can't seem to get parameters passed correctly
 - Might have to “wrap” the apps using AppleScript or Automator to create “runnable apps”
 - Might be able to solve using & on end of Terminal command
 - `/Applications/Jamulus1.app/Contents/MacOS/Jamulus --clientname Keyboards --inifile Documents/Jamulus/Keyboards.ini`
 - `/Applications/Jamulus2.app/Contents/MacOS/Jamulus --clientname Drums --inifile Documents/Jamulus/Drums.ini`
- JamulusKeys > Settings
 - Device set to in:iConnectAUDIO4+/out:iConnectAUDIO4+
 - Input Channel Mapping
 - L set to Channel 1 (keyboards)
 - R set to Channel 2 (keyboardist's mic)
 - Output Channel Mapping
 - L set to AI output 2 (keyboardist's headphones)
 - R set to AI output 2 (keyboardist's headphones)
 - Enable Small Network Buffers is unchecked
 - Buffer Delay set to 5.33ms
 - Jitter Buffer > Auto is checked
 - Misc > Audio Channels set to Mono
 - Misc > Audio Quality set to Normal
 - Misc > New Client Level set to 50%

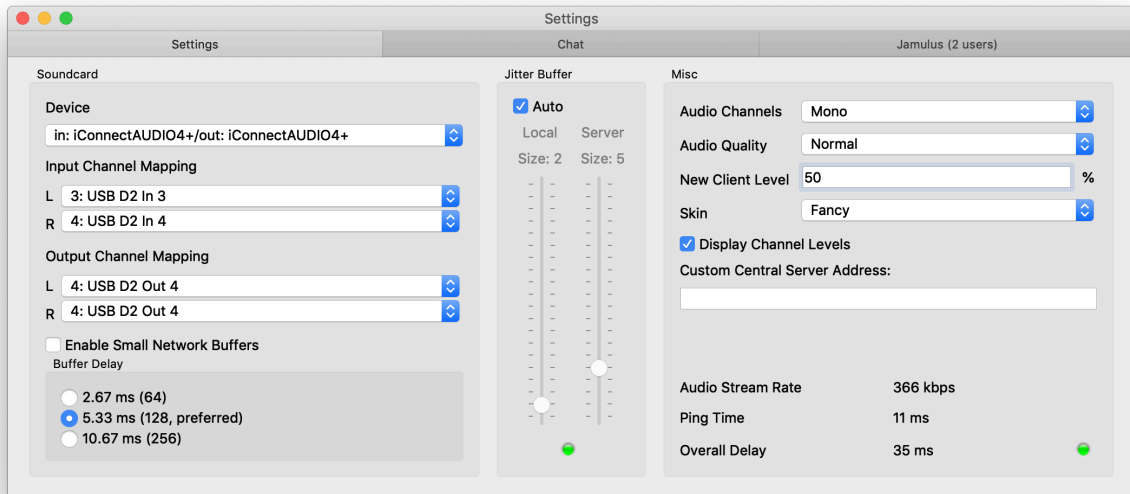


- JamulusKeys > View > My Profile
 - Name set to keyboardist's name
 - Instrument set to Keyboard
- JamulusKeys > Connect
 - Adjust mixer levels for what the keyboardist wants to hear

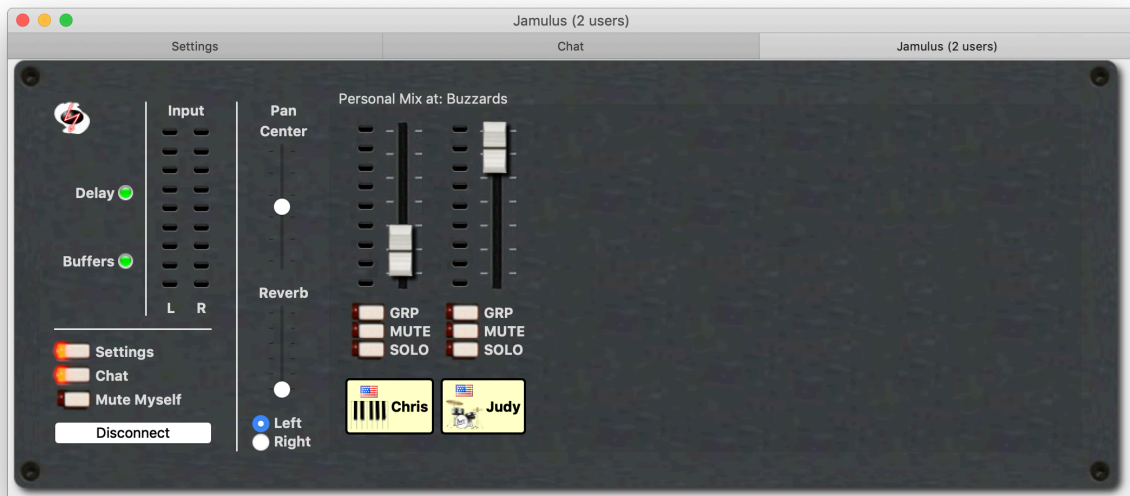


- JamulusDrums > Settings
 - Device set to in:iConnectAUDIO4+/out:iConnectAUDIO4+
 - Input Channel Mapping
 - L set to Channel 3 (drums)
 - R set to Channel 4 (drummer's mic)
 - Output Channel Mapping

- L set to AI output 4 (drummer's headphones)
- R set to AI output 4 (drummer's headphones)
- Enable Small Network Buffers is unchecked
- Buffer Delay set to 5.33ms
- Jitter Buffer > Auto is checked
- Misc > Audio Channels set to Mono
- Misc > Audio Quality set to Normal
- Misc > New Client Level set to 50%



- JamulusDrums > View > My Profile
 - Name set to drummer's name
 - Instrument set to Drum Set
- JamulusDrums > Connect
 - Adjust mixer levels for what the drummer wants to hear



- After Session
 - Quit JamulusKeys
 - Quit JamulusDrums
 - iConfig > File > Save Preset > "local"
 - Unmutes analog inputs that were muted, so they play through monitor speaker
 - MacOS > System Preferences > Sound
 - Output set to system default (usually Internal Speakers)
 - Output volume set to 50% and not Muted
 - Input set to system default (usually Internal Microphone)
 - Quit System Preferences

My Config - Play Recording In Jamulus

When my band is learning a new song, we'll sometimes play along with the original recording, which can often be found on YouTube or a streaming music service (iTunes, Spotify). I route the audio to Jamulus by running an additional instance of Jamulus.

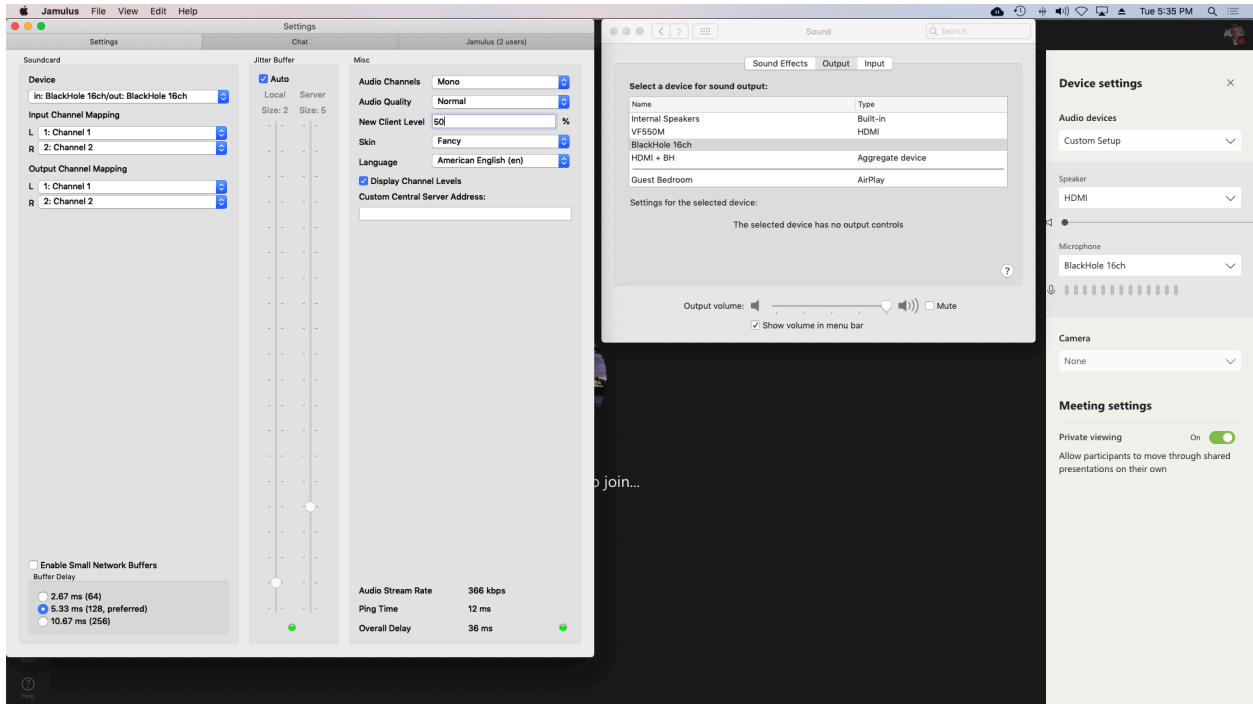
- One-Time Setup
 - Finder > Applications > Utilities > Audio MIDI Setup > Show Audio Devices
 - Ensure that every listed device (particularly BlackHole) is set to 48KHz
 - Finder > Applications
 - Right-click Jamulus, then Duplicate
 - Rename Jamulus as JamulusSong
- Every Session
 - MacOS > System Preferences > Sound
 - Output set to BlackHole
 - Output volume set to 100%
 - Quit System Preferences
 - JamulusSong > Settings
 - Device set to in:System/out:System
 - JamulusSong > View > My Profile
 - Name set to "Song"
 - Instrument set to Recorder (but can be anything)
 - JamulusSong > Connect
 - Adjust mixer level for "Song" to 0%
 - Audio application (iTunes, Spotify, YouTube, etc.)
 - Play music
 - Adjust volume while viewing Jamulus mixer output level, aiming for one yellow LED
- After Session
 - Quit JamulusSong
 - MacOS > System Preferences > Sound
 - Output set to system default (usually Internal Speakers)
 - Output volume set to 50% and not Muted
 - Quit System Preferences

My Config - Using Microsoft Teams

My employer uses Microsoft Teams, a web conferencing and text chat application. For a recent event, some co-workers and I rehearsed a few songs via Jamulus, then performed for the company via a Teams conference. I routed the Jamulus output to Teams, but did not attempt to route the Teams output back to Jamulus, since that could have created a feedback loop. Instead, I ran the Teams application on a separate device, with the microphone and speaker

muted, then unmuted the speaker between songs so that I could hear the audience's questions and comments.

- On my primary Mac
 - I configured Jamulus as usual and connected to my cloud server
- On a second Mac
 - I installed BlackHole
 - Finder > Applications > Utilities > Audio MIDI Setup > Show Audio Devices
 - Ensure that every listed device (particularly BlackHole) is set to 48KHz
 - System Preferences > Sound
 - Input = BlackHole
 - Output = BlackHole
 - Output Volume = 100%
 - I installed and configured Jamulus
 - Device set to in:BlackHole/out:BlackHole
 - Input Channel Mapping
 - L set to Channel 1
 - R set to Channel 2
 - Output Channel Mapping
 - L set to Channel 1
 - R set to Channel 2
 - Enable Small Network Buffers is unchecked
 - Buffer Delay set to 5.33ms
 - Jitter Buffer > Auto is checked
 - Misc > Audio Channels set to Mono
 - Misc > Audio Quality set to Normal
 - Misc > New Client Level set to 50%
 - I connected Jamulus to my cloud server
 - Note that the Jamulus mixer settings affect the output to Teams, so I left them all at 50%
 - I installed and configured Teams
 - Speaker = HDMI (but this could be internal speaker or any choice but BlackHole)
 - Microphone = BlackHole
 - In Teams, I joined the web conference
- On an iPad (although it could be any device with a Teams application)
 - I installed Teams and joined the web conference
 - I muted my microphone
 - I muted my speaker

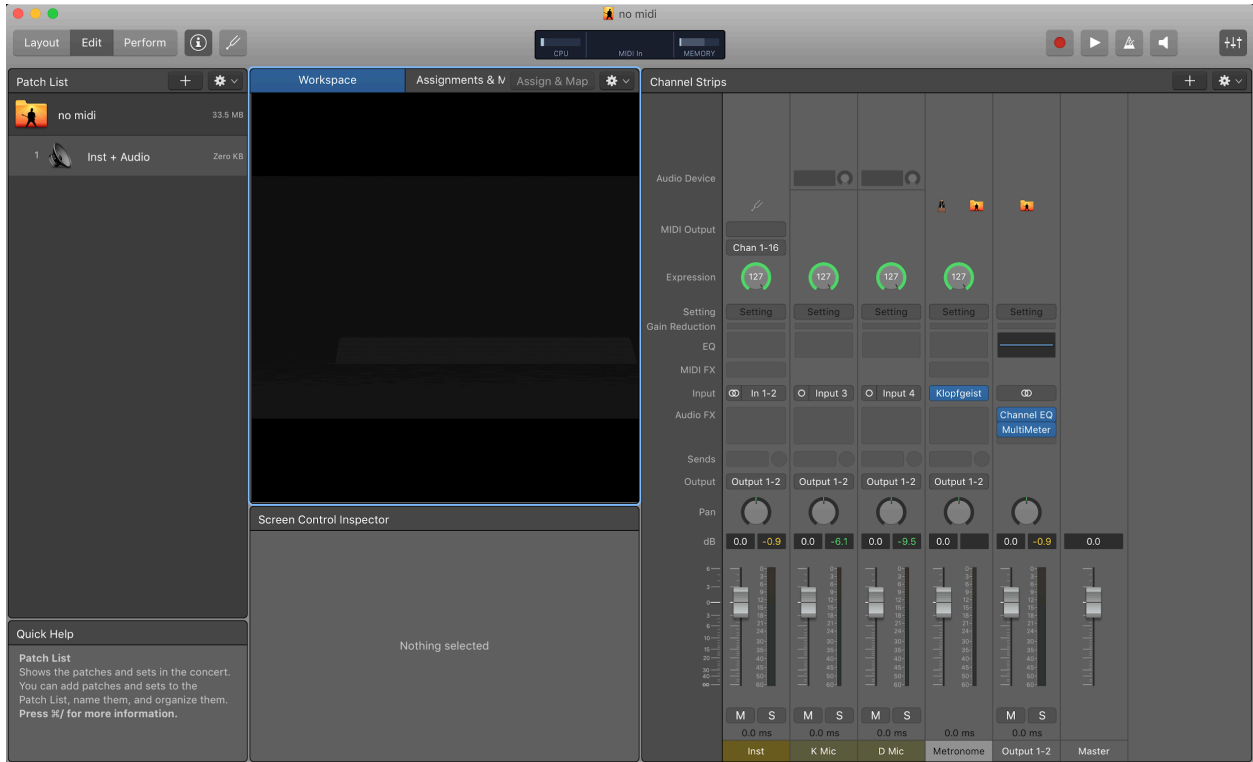


My Config - Using DAW & Virtual Audio Driver

How I have Jamulus configured for my [full equipment stack](#) when using 1 Jamulus instance and merging more than 2 inputs via a [DAW](#) and [Virtual Audio Driver](#). I need to review and update this section.

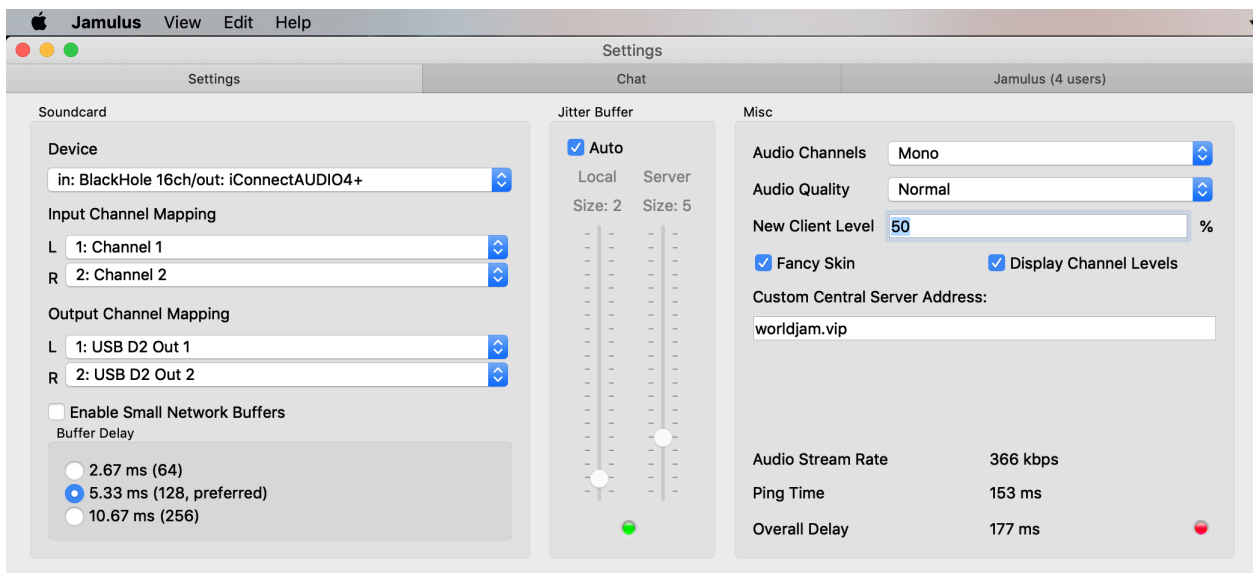
- [Diagram of my equipment layout and cabling](#)
- One-Time Setup
 - Finder > Applications > Utilities > Audio MIDI Setup > Show Audio Devices
 - Ensure that every listed device (particularly BlackHole) is set to 48KHz
 - [maybe not needed] Auracle > iConnectivity AUDIO4+ > Audio Mixer
 - Direct Monitor Destination set to Off for Headphones
 - Quit Auracle
 - MainStage > Layout Mode
 - Empty (delete all controls)
 - MainStage > Edit Mode
 - New Set
 - New Patch
 - Add External Instrument Channel Strip
 - External Instrument Channel Strips don't allow a single input (mono), so the instruments are merged to a single channel strip
 - Input set to Input 1-2 (keyboards and drums)
 - Output set to Output 1-2

- Enable Small Network Buffers is unchecked
- Buffer Delay set to 5.33ms
- Jitter Buffer > Auto is checked
- Misc > Audio Channels set to Mono
- Misc > Audio Quality set to Normal
- Misc > New Client Level set to 50%



Channel Strips

Audio Device						
MIDI Output						
Expression	127	127	127	127		
Setting	Setting	Setting	Setting	Setting	Setting	
Gain Reduction						
EQ						
MIDI FX						
Input	In 1-2	Input 3	Input 4	Klopfgeist		
Audio FX					Channel EQ MultiMeter	
Sends						
Output	Output 1-2	Output 1-2	Output 1-2	Output 1-2		
Pan						
dB	0.0 -0.9	0.0 -6.1	0.0 -9.5	0.0	0.0 -0.9	0.0
	M S	M S	M S		M S	
	0.0 ms	0.0 ms	0.0 ms	0.0 ms	0.0 ms	
	Inst	K Mic	D Mic	Metronome	Output 1-2	Master



- Every Session
 - MacOS > System Preferences > Sound
 - Output set to system default (usually Internal Speakers)
 - Output volume set to Mute
 - Input set to system default (usually Internal Microphone)
 - Quit System Preferences
 - iConfig > File > Open Preset > (previously saved configuration file)

- Quit iConfig
 - MainStage > Open > (previously saved configuration file)
 - MainStage > Preferences > Audio
 - Audio Output set to BlackHole
 - Audio Input set to iConnectivity AUDIO4+
 - Sample Rate set to 48KHz
 - Jamulus > Settings
 - Should default to same settings, but adjust if not
 - Jamulus > Connect
- After Session
 - Quit Jamulus2
 - Quit Jamulus1
 - MacOS > System Preferences > Sound
 - Output set to system default (usually Internal Speakers)
 - Output volume set to 50% and not Muted
 - Input set to system default (usually Internal Microphone)
 - Quit System Preferences

My Server Usage

How I'm using Jamulus servers.

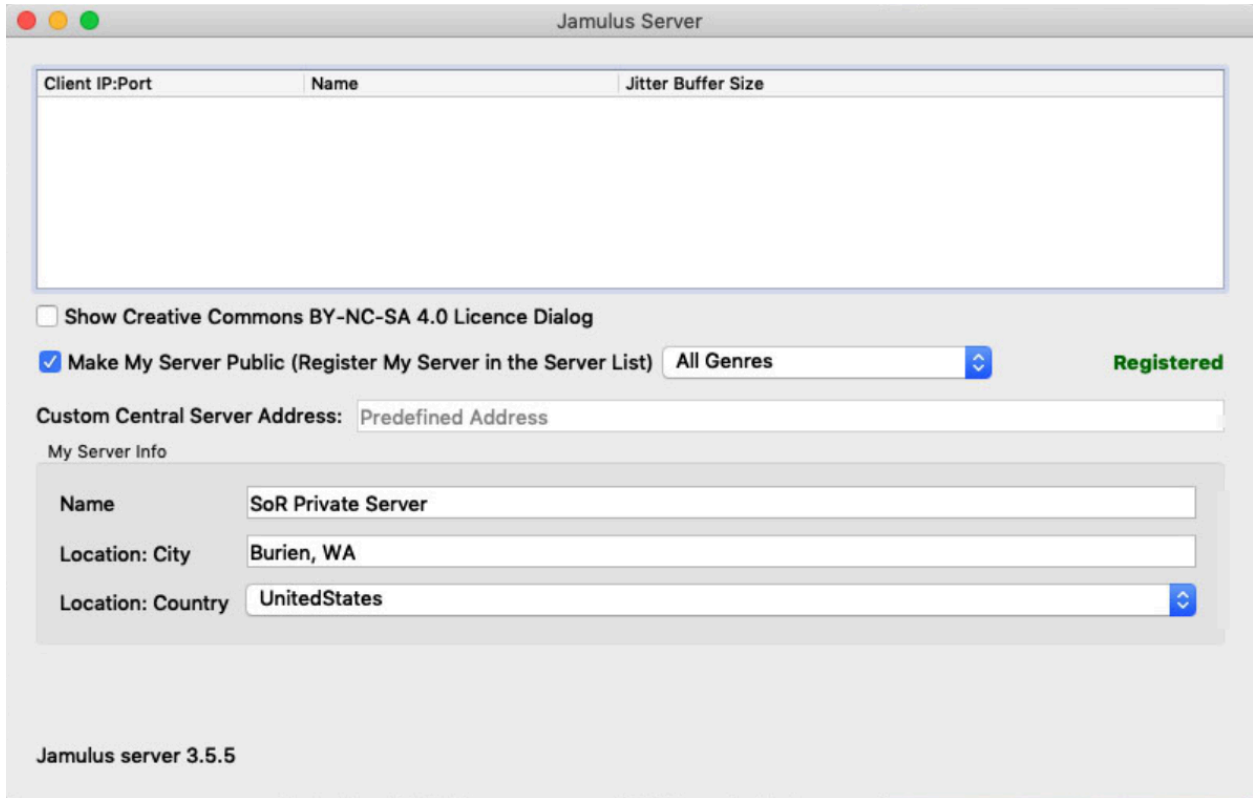
- Local server
 - I've experimented with running a Jamulus Server on a second Mac, which has a hard-wired ethernet connection (not WiFi), but have abandoned it in favor of a cloud server
 - I haven't tried to make the server private yet and have not made any adjustments to port forwarding, etc.
 - My hope is that, by running the server as physically close to the drums as possible, I can reduce latency for the drummer, making it easier for the drummer to keep time
 - When connected to the server, in Jamulus I'm seeing 1-5ms Ping Time and 20ms Overall Delay
 - My band members are seeing Overall Delay of 20-70ms and we seem to get a better balance of Overall Delay (everyone at 40ms or below) by connecting to a cloud server rather than my local server - it's likely that my cable modem and router are adding latency that we're not experiencing with cloud servers, and/or that my internet server provider has poor routing into my home that is adding latency.
- Cloud servers
 - I live in the Seattle, WA, USA area and there are nearby cloud servers running on [Microsoft Azure](#), [Google Cloud Platform](#) (GCP), and [Amazon Web Services](#) (AWS)

- The nearest Azure data center is in eastern WA, so 100+ miles from my home - in Jamulus, I'm seeing 15-20ms Ping Time and 40ms Overall Delay
- The nearest GCP and AWS data centers are in OR, 150+ miles from my home - in Jamulus, I'm seeing 20-30ms Ping Time and 50ms Overall Delay
- There is a hosted-at-home server within 20 miles of my house, running on a high-powered Mac, with 15ms Ping Time and 40ms Overall Delay - the owner has fiber optic internet (CenturyLink), which appears to meaningfully improve Ping Time and Overall Delay over cable internet (Comcast), such as my local server above
- There is a cloud server running on [Vultr](#), which has a data center in Seattle and within 10 miles of my home, with 10ms Ping Time and 25ms Overall Delay - this appeared to be my lowest latency option, and I set up [my own Vultr server](#) for my band to use for rehearsals
- Ultimately, I decided to set up my own cloud server (configuration below)

My Server Config - Local

How I have Jamulus Server configured, when I choose to run a local server.

- Jamulus Server > Show Creative Commons License Dialog is unchecked
- Jamulus Server > Make My Server Public is checked
- Jamulus Server > Genre is set to All Genres
- Jamulus Server > My Server Info > Name is (a name that I've chosen)
- Jamulus Server > My Server Info > City is (my location)
- Jamulus Server > My Server Info > Country is (my location)



When I want to record band rehearsals, I start Jamulus Server from a Terminal command line

- Although a command line can be copy/pasted to Terminal, be aware that Terminal does not support “smart quotes”, so any double quotes likely need to be typed (not pasted)
- `/Applications/Jamulus.app/Contents/MacOS/Jamulus --server --recording Downloads --welcomemessage "This server is for private band rehearsals. If it's up, we're probably rehearsing. You're welcome to listen, but please don't interrupt. Thanks."`

My Server Config - Cloud (Debian)

How I have configured my Jamulus cloud server.

- I followed Andrew Ferguson's [detailed instructions for installing Jamulus Server on Vultr](#) but modified them to install the latest release version (not the development version) and named my server “jamulus” - specific changes are below
- After creating a Vultr account and High Frequency Compute server (\$6/month at the time of this writing) running Debian 10 (not 11, which doesn't work with these instructions), I selected the server under Products and chose Overview to see the username (“root”), password (randomly assigned by Vultr.com), and IP address
- I opened Mac Terminal and used “ssh root@myipaddress” to connect to the server - ensure that you have the window size large enough to avoid vertical and horizontal scrolling, if possible
- On the server, execute these commands

- Update Linux
 - apt-get update
 - apt-get -y dist-upgrade
- Install and configure UFW firewall
 - apt-get install ufw
 - ufw default deny incoming
 - ufw default allow outgoing
 - ufw allow ssh
 - ufw allow 22124/udp
 - ufw enable
- Install dependencies
 - apt-get -y install git build-essential qtdeclarative5-dev qt5-default qttools5-dev-tools libjack-jackd2-dev
- Download Jamulus source code
 - cd /tmp/
 - git clone https://github.com/jamulussoftware/jamulus.git
- Compile source code
 - cd jamulus
 - git checkout latest
 - qmake "CONFIG+=nosound headless" Jamulus.pro
 - make clean
 - make
 - make install
 - mv Jamulus /usr/local/bin/
- Create a user
 - adduser --system --no-create-home jamulus
- Create a directory in which to store recordings
 - mkdir -p /var/jamulus/recording
 - chown jamulus /var/jamulus/recording
- Create a systemd unit file
 - nano /etc/systemd/system/jamulus.service
- Paste this content into the systemd unit file and save it
 - [Unit]
 - Description=Jamulus-Server
 - After=network.target
 -
 - [Service]
 - Type=simple
 - User=jamulus
 - Group=nogroup
 - NoNewPrivileges=true
 - ProtectSystem=true
 - ProtectHome=true

- Nice=-20
 - IOSchedulingClass=realtime
 - IOSchedulingPriority=0
 -
 - ##### Change this to set genre, location and other parameters.
 - ##### See <https://jamulus.io/wiki/Command-Line-Options>
 - #####
 - ExecStart=/usr/local/bin/Jamulus \
 - server \
 - nogui \
 - recording /var/jamulus/recording/ \
 - directoryserver anygenre2.jamulus.io:22224 \
 - serverinfo "My Server;Seattle, WA;225" \
 - welcomemessage "Welcome to my server" \
 - license
 -
 - Restart=on-failure
 - RestartSec=30
 - StandardOutput=journal
 - StandardError=inherit
 - SyslogIdentifier=jamulus
 -
 - [Install]
 - WantedBy=multi-user.target
- Set permissions for the unit file
 - `chmod 644 /etc/systemd/system/jamulus.service`
- Start Jamulus and verify that it's running
 - `systemctl start jamulus`
 - `systemctl status jamulus`
- When creating the jamulus.service file using nano, I adjusted some command line parameters, so my ExecStart line is:
 - ExecStart=/usr/local/bin/Jamulus \
 - server \
 - nogui \
 - serverinfo "School of Rock WS;Seattle, WA;225" \
 - welcomemessage "<h1>School of Rock West Seattle Jamulus Server</h1><p>This server is primarily for School of Rock West Seattle students, but is also open for public use. It is often busy with band rehearsals on Wednesdays, Fridays, and Sundays 6-9pm PST (UTC-8). It is hosted at Vultr.com in Seattle, WA, USA by Chris Rimple. Please contact chrisrimple@yahoo.com with

questions.

For general information about Jamulus and Jamulus resources, see the Remote Band Rehearsals document:

https://docs.google.com/document/d/1smcvsxdaaViPQvGMQHmah_6BQeqowhmGSFMHfnlY2FI/

```

--numchannels 20 \
--multithreading \
--fastupdate \
--directoryserver anygenre2.jamulus.io:22224

```

My server is usually public, but when I want it to be private to prevent possible interruptions, I edit the jamulus.service file to remove the directory server parameter:

- I open Mac Terminal and use “ssh myservername@myipaddress” to connect to the server
 - nano /etc/systemd/system/jamulus.service
 - Delete the --directoryserver parameter
 - systemctl daemon-reload
 - service jamulus restart

Prior to my band’s rehearsals, I reboot the server:

- On Vultr, under Products > Overview, I select the server and choose Restart
- I open Mac Terminal and use “ssh myservername@myipaddress” to connect to the server
 - systemctl start jamulus
 - systemctl status jamulus

When my band is rehearsing and I want to record, I edit the jamulus.service file to add the recording parameter to the end:

- I open Mac Terminal and use “ssh myservername@myipaddress” to connect to the server
 - nano /etc/systemd/system/jamulus.service
 - ExecStart=/usr/local/bin/Jamulus \
--server \
--nogui \
--serverinfo "School of Rock WS;Seattle, WA;225" \
--welcomemessage "<h1>School of Rock West Seattle Jamulus Server</h1><p>This server is primarily for School of Rock West Seattle students, but is also open for public use. It is often busy with band rehearsals on Wednesdays, Fridays, and Sundays 6-9pm PST (UTC-8). Recordings are enabled on this server

and are automatically deleted after 24 hours. It is hosted at Vultr.com in Seattle, WA, USA by Chris Rimple. Please contact chrisrimple@yahoo.com with questions. For general information about Jamulus and Jamulus resources, see the Remote Band Rehearsals document: https://docs.google.com/document/d/1smcvsxdaaViPQvGMQHmah_6BQeqowhmGSFMHfnly2FI/

```

--numchannels 20 \
--multithreading \
--fastupdate \
--directoryserver anygenre2.jamulus.io:22224 \
--recording /var/jamulus/recording/

```

- `systemctl daemon-reload`
- `service jamulus restart`

Lately, I've been leaving recording enabled on the server at all times:

- I've enabled a cron job, running at 12PM daily (server local time), to delete recordings that are more than 24 hours old
 - `cd /etc`
 - `nano crontab`

```

■ 00 12 * * * root find
/var/jamulus/recording/* -mtime +1 -type d -exec rm
-rf {} \;

```
- When a band rehearsal starts, after all the musicians have joined the server, I force recording to restart so that the recording files have file names based on the Jamulus names of the musicians
 - `systemctl kill -s SIGUSR1 jamulus`
- When a band rehearsal ends, I force recording to restart so that the recording files are closed, allowing me to download them
 - `systemctl kill -s SIGUSR1 jamulus`
- When I want to switch the recorder on or off
 - `systemctl kill -s SIGUSR2 jamulus`

To update the server to a new version of Jamulus:

- Read the [Jamulus Change Log](#) to see if any command line parameters have been changed or removed - the ExecStart line in the `jamulus.service` file may need to be updated
- I open Mac Terminal and use "`ssh myservername@myipaddress`" to connect to the server
 - `cd /tmp/`

- git clone https://github.com/jamulussoftware/jamulus.git
- cd jamulus
- git checkout latest
- qmake "CONFIG+=nosound headless" Jamulus.pro
- make clean
- make
- make install
- mv Jamulus /usr/local/bin/
- Then I reboot the server (above)

To downgrade the server to an older version of Jamulus, replace “r3_8_0” with the desired version:

- I open Mac Terminal and use “ssh myservername@myipaddress” to connect to the server
 - cd /tmp/
 - git clone https://github.com/jamulussoftware/jamulus.git
 - cd jamulus
 - git checkout r3_8_0
 - qmake "CONFIG+=nosound headless" Jamulus.pro
 - make clean
 - make
 - make install
 - mv Jamulus /usr/local/bin/
- Then I reboot the server (above)

My Server Config - Cloud (Ubuntu)

How I have configured my Jamulus cloud server.

- I followed Simon Tomlinson’s [detailed instructions for installing Jamulus Server on AWS](#), but did it on Vultr.com instead
- After creating a Vultr account and High Frequency Compute server (\$6/month at the time of this writing), I selected the server under Products and chose Overview to see the username (“root”), password (randomly assigned by Vultr.com), and IP address
- I opened Mac Terminal and used “ssh myservername@myipaddress” to connect to the server - ensure that you have the window size large enough to avoid vertical and horizontal scrolling, if possible
- I upgraded to the latest Ubuntu
 - apt-get update
 - apt-get dist-upgrade
- I installed using Simon’s script, which took a few minutes to complete
- Afterward, I edited the /etc/systemd/system/jamulus.service file using nano and adjusted some command line parameters - see my Debian configuration above for examples

- I did not need to configure Jamulus to auto-start if the server reboots - this is done by Simon's script
- I followed the [Vultr instructions to install an FTP server](#), so that I could download recordings

My server is usually public, but when I want it to be private to prevent possible interruptions, I edit the jamulus.service file to remove the central server parameter - see my Debian configuration above for examples.

Prior to my band's rehearsals, I reboot the server - see my Debian configuration above for examples.

When my band is rehearsing and I want to record, I don't have to change anything, because recording is always enabled (see ExecStart parameters above). Because recording is enabled at all times, I've enabled a cron job, running at 12pm daily (server local time), to delete recordings that are more than 24 hours old - see my Debian configuration above for examples.

To update the server to a new version of Jamulus:

- Read the [Jamulus Change Log](#) to see if any command line parameters have been changed or removed - the ExecStart line in the jamulus.service file may need to be updated
- Follow the instructions in Simon Tomlinson's [detailed instructions for installing Jamulus Server on AWS](#), specifically:
 - `wget https://worldjam.vip/jamulus/setup-jamulus.sh`
 - `chmod 744 setup-jamulus.sh`
 - `./setup-jamulus.sh`
 - When asked if you want to set up a new server, respond "N" to upgrade the existing server
- Then I reboot the server (above)

My Recordings

With recording enabled on my Jamulus server, and having administrative access, I download recordings using [Cyberduck](#) (although any application supporting SFTP download will work). How I've configured Cyberduck:

- In Cyberduck, create a new Bookmark
- Set the Protocol to SFTP
- Set the Nickname to "SoR WS Jamulus Server"
- Set the Server to the IP address of my Jamulus server (which I obtain from Vultr.com)
- Set the Username to "root"
- Set the Password to the password of my Jamulus server (which I obtain from Vultr.com)
- Set the Path to `"/var/jamulus/recording"`

After a Jamulus session has completed, I run Cyberduck and connect to that Bookmark. Cyberduck displays a list of list recording folders - I select the most recent and Download.

When the download is complete, I open the recording using Audacity, where I can trim the beginning and end, save specific sections of the Jamulus session, save individual performers, and so on.

JamKazam

JamKazam is similar to Jamulus. Some people have said that it used to be good, but now the servers are overloaded and it crashes regularly. It doesn't show latency values when looking for a server to join. I've spent a few hours trying it, but prefer Jamulus.

General Information & FAQ

- Available for Windows, Mac, and Linux
- <https://www.jamkazam.com>
- <https://forum.jamkazam.com>

Installation

- <https://www.jamkazam.com/downloads>
- Saul Richardson has [a short blog post with configuration instructions](#)
- The JamKazam forum has [some information on using it with a DAW](#)

Instructional Videos

- <https://www.youtube.com/watch?v=-wO3532TMdY>
- <https://www.youtube.com/watch?v=VgP5BkGbnMY> - route audio to OBS

Usage Videos

- <https://www.youtube.com/watch?v=I-5zCQs8A6I>

Social Media & Forums

- Facebook page [JamKazam](#)
- Facebook page [JamKazam Users](#)
- Facebook page [JamKazam Live Jams](#)
- Facebook page [JamKazam Music Festival](#)
- Facebook group [JamKazam Jammers](#)
- Facebook group [JamKazam - The Global Jam Session](#)

SoundJack

SoundJack uses the peer-to-peer model. I haven't tested it yet, but the sample videos seem promising.

General Information & FAQ

- Available for Windows, Mac, and Linux
- Requires 48KHz output (see note in [AI](#))
- <https://www.soundjack.eu>
- <https://www.soundjack.eu/index.php/problems>

Installation

- <https://www.soundjack.eu/index.php/howto>

Instructional Videos

- <https://www.soundjack.eu/index.php/howto>

Usage Videos

- https://www.youtube.com/watch?time_continue=1449&v=67FsQYHomYg
- <https://www.soundjack.eu/index.php/demovideos>

Social Media & Forums

- Facebook page [SoundJack](#)
- Facebook group [SoundJack](#)

JackTrip

JackTrip uses the peer-to-peer model. I haven't tested it yet, but setup and configuration is all done via command line, so not a good choice for anyone that isn't significantly tech skilled. It seems to be good for 2 musicians, but may be difficult to get working with 3 or more. Some people have reported that it crashes frequently.

General Information & FAQ

- Available for Windows, Mac, and Linux
- <https://www.jacktrip.org>
- <https://ccrma.stanford.edu/software/jacktrip/>
- <https://ccrma.stanford.edu/groups/soundwire/software/jacktrip/>

Installation

- <https://mct-master.github.io/portal/2020/03/31/jacktrip.html>
- <http://mdessen.com/wp-content/uploads/2013/01/jacktripBasics1.pdf>
- <https://www.haven2.com/index.php/cookbook-for-an-online-jamming-session>

Instructional Videos

- <https://www.youtube.com/watch?v=-dWTZTv7bUY>
- https://www.youtube.com/playlist?list=PL9rgN0ig4hMQ8Pf3YnV_mrpqslvFL07X
- https://www.youtube.com/watch?v=3HP_Z6mg7QU

- <https://www.youtube.com/watch?v=H6D9DSQ-h9c>

Usage Videos

- <https://www.youtube.com/watch?v=m9S3ITVnL90>

Social Media & Forums

- None

Delayed Sync Applications

Delayed Sync applications are great for unstructured jamming, “noodling” on an instrument, and learning to improvise, but not good for playing through an entire song end-to-end or as a replacement for in-person band/chorus rehearsals. These applications solve latency by increasing it, from hundredths of a second to multiple measures. To quote the Ninjam documentation:

NINJAM provides a solution by making latency much longer. Latency in NINJAM is measured in measures... The NINJAM client records and streams synchronized intervals of music between participants. Just as the interval finishes recording, it begins playing on everyone else’s client. So when you play through an interval, you’re playing along with the previous interval of everybody else, and they’re playing along with your previous interval.

Think of it like this: the drummer counts in and plays 8 measures of a song, then everyone else hears the count in and plays along - meanwhile, the drummer is always 8 measures ahead and hears everyone else as 8 measures behind. Yes, the drummer could repeat the first 8 measures to be “in sync” with everyone else, but then it gets worse: all band members are playing the first 8 measures together, but then they hear what everyone played on measure 1 when they are ready to play measure 9.

Where Delayed Sync applications can be valuable is when playing with a short, looping backing track or playing the verse-chorus portion of a song repeatedly. For example, the song “Ghostbusters” has a repetitive verse and chorus pattern of 8m intro, 16m verse, 16m chorus - playing that pattern repeatedly on drums and bass can make it easy for other instruments to improvise “on top”, but they won’t be able to play the full song from beginning to end.

Ninjam / Ninbot / Reaper

Ninjam is not an application, but rather a set of servers that are accessed using an application like Reaper, Jamtaba, and others. Ninbot is a subset of the public servers, run by a group of Ninjam fans. I’ve used Ninjam a few times and enjoy it, but recommend the [Jamtaba](#) user interface for beginners, since Reaper is a [DAW](#) with much more capability and complexity.

General Information & FAQ

- Available for Windows, Mac, and Linux
- https://www.reddit.com/r/Guitar/comments/butxrs/discussion_online_jamming_everything_you_ever/
- <http://ninbot.com/getstarted>
- <http://ninbot.com/content/about-ninbot/>
- <http://ninbot.com/content/faq>
- <https://www.reaper.fm>
- <https://www.reaper.fm/userguide.php>

Installation

- <https://www.reaper.fm/download.php>

Instructional Videos

- <https://www.youtube.com/watch?v=8sew613kSdA>
- https://www.youtube.com/watch?v=EK7XGoV7F_U
- <https://www.youtube.com/watch?v=P-cUmSw7nF4>

Usage Videos

- <https://www.youtube.com/watch?v=flAQmQnInVI>
- https://www.youtube.com/watch?v=-s70M9mov_Q
- <https://www.youtube.com/watch?v=sgo4xDeU9W0>

Server Installation

- <https://www.cockos.com/ninjam/server-guide.php>

Server Installation Videos

- <https://www.youtube.com/watch?v=9sY2P41gIMA&>
- <https://www.youtube.com/watch?v=WwT6hl8Z3Cl>
- <https://www.youtube.com/watch?v=yNc383Q2pUs>

Social Media & Forums

- Facebook group [Ninjam](#)
- Facebook page [Reaper DAW](#)
- Facebook group [Reaper User Group](#)
- Facebook group [Reaper Blog Community](#)
- Facebook group [Reaper DAW Tips & Tricks](#)
- Facebook group [Reaper DAW Users](#)

Jamtaba

Jamtaba provides a simple interface to Ninjam servers, including a map showing the location of users accessing each server, which can help in finding other musicians that speak the same language. It also provides the ability to “listen in” to a server before joining it, to see if the genre/style of music being played is of interest to you, or just to hear what others are doing with Ninjam.

Jamtaba is limited to 2 input channels with a sub-channel for each - if you need more inputs, you'll have to use a [DAW](#) to merge those inputs to a single output before sending it to Jamtaba.

General Information & FAQ

- Available for Windows, Mac, and Linux
- <https://jamtaba-music-web-site.appspot.com>
- <https://github.com/elieserdejesus/JamTaba/wiki/f.a.q>
- <https://github.com/elieserdejesus/JamTaba/wiki/Jamtaba%27s-user-guide>

Installation

- <https://github.com/elieserdejesus/JamTaba/wiki/Installation>

Instructional Videos

- <https://www.youtube.com/watch?v=rzwyUkrBKNM>
- <https://www.youtube.com/watch?v=pnZzA7ovAOU>
-

Usage Videos

- <https://www.youtube.com/watch?v=DQoGa-35r1Q>
- <https://www.youtube.com/watch?v=4XP1chq3COc>
- <https://www.youtube.com/watch?v=sVIUoQJr9PA>

Social Media & Forums

- None

My Configuration - Full Equipment Stack

How I have Jamtaba configured for my [full equipment stack](#).

- Every Session
 - Jamtaba
 - 1 channel for AI mono channel 1 (keyboards)
 - Sub-channel for AI mono channel 3 (keyboardist's mic)
 - 1 channel for AI mono channel 2 (drums)
 - Sub-channel for AI mono channel 4 (drummer's mic)

- Any [loops](#) loaded in the keyboards channel

My Configuration - MIDI Keytar

How I have Jamtaba configured for my MIDI keytar, which requires a [DAW](#) for sound generation.

- One-Time Setup
 - Mac Finder > Applications > Utilities > Audio MIDI Setup
 - Window > Show Audio Devices
 - Create Multi-Output Device
 - Select Built-in Microphone, Built-in Output, and BlackHole
 - Set Clock Source to BlackHole
 - Set Sample Rate to 48KHz
 - Enable Drift Correction for all
- Every Session
 - MacOS > Preferences > Sound
 - Output set to Headphones with 25% Input Volume
 - Input set to External Microphone
 - MainStage > Preferences > Audio
 - Audio Output set to BlackHole
 - Audio Input set to Automatic
 - Sample Rate set to 48KHz
 - Jamtaba > Preferences > Audio
 - Audio Device Input set to Multi-Output Device
 - Audio Device Output set to Built-in Output
 - Sample Rate set to 48000
 - Jamtaba
 - 1 channel for keytar and microphone

Jammr.net

Seems to be a solution similar to Ninjam / Reaper, but I haven't tested it yet.

Loops

My Loops

Using GarageBand, I've created [a variety of loops](#) for use with Ninjam/Reaper/Jamtaba, that could also be used with other applications. I've [shared the GarageBand files, allowing anyone to download and edit them, as well as MP3s](#). The filename for each loop includes the BPM,

length in measures, key signature, and chord progression, as well as a loop name. Drum-only loop filenames omit the key signature and chord progression.

Jamtaba

In Jamtaba, loop “layers” are limited to 64 beats, which is 16 measures in 4/4 time. To play a loop that is longer than 64 beats, it must be separated into parts, one per layer. All of [My Loops](#) that are longer than 64 beats have already been separated into parts for use in layers.

Assuming a 4/4 time signature, BPI (beats per interval) should be set to “number of measures times 4”. For example, for a 16 measure loop in 4/4 time, set BPI = 64. In 4/4 time, 4m = 16BPI, 8m = 32BPI, 12m = 48BPI, and 16m = 64BPI.

Start playback of the loop layer(s) at any time - it will automatically sync to the appropriate spot in the BPI countdown.

GarageBand / Logic / Other DAWs

In GarageBand and Logic, import a loop MP3, then enable looping and set the looping region to match the length of the MP3. Start playback of the MP3 and ensure that output from the DAW is routed to the remote band rehearsal application.

WikiLoops

[WikiLoops](#) is a community of musicians that record loops and share them, allowing others to use them freely for practice, to record with, etc. Most loops are full-length songs, not short loops like those usually used with remote band rehearsal applications, and are not free. If you’re looking for other musicians with which to do remote multi-track recording, this may be an option.

[END]