

QuaxTrip Manual

QuaxTrip is a set of Max patches that makes low-latency uncompressed audio and messaging interconnections over the internet, intended for musicians wishing to play together remotely.

Background

QuaxTrip runs Miller Puckette's Quacktrip Pure Data patch within Cycling '74's Max. Quacktrip, in turn, is an implementation, in Pure Data, of Chris Chafe's Jacktrip network protocol, based on jacktrip.pd by Roman Haefeli and Johannes Schuett. It establishes a low-latency, point-to-point connection between two sites, with no audio compression.

Installation

1. If you don't already have Max 8, [download](#) and install it. Note that you do not need to purchase Max to use QuaxTrip, it will run in the unregistered version.
2. [Download](#) and install Pure Data (version 0.51-2 or later).
3. If Max is running, quit Max.
4. [Download](#) PdMax (version 0.54 or greater), unzip, and place in the ~/Documents/Max 8/library directory.
5. [Download](#) the latest release of QuaxTrip, unzip, and place in the ~/Documents/Max 8/library directory.

Quick Start

There are two versions of QuaxTrip: **QuaxTrip Duet.maxpat** and **QuaxTrip Quintet.maxpat**. The only difference between these two, as the names imply, is that **Duet** allows only one remote partner while **Quintet** allows up to four. If you are playing with only one partner, **Duet** frees up some screen space and uses somewhat less CPU.

*Note: It is **highly** recommended that you are connected to the internet via Ethernet rather than WiFi. In a pinch, WiFi might work, but you are much more likely to get dropouts.*

How to play with a remote partner:

1. Both you and your remote partner launch **QuaxTrip Duet.maxpat**.
2. If this is the first time you've launched QuaxTrip, you will be presented with a dialog box instructing you to locate and select the Pd app (binary) on your computer. You will only need to do this the first time you run the patch.
 - a. macOS: The Pd app is typically in the /Applications directory.

- b. Windows: The Pd app is typically in the /Program Files/Pd/bin directory.
3. In the **Local** channel (top left corner of the patch):
 - a. Set the **Input Starting Channel** to reflect the first channel (or only channel, if mono) of incoming audio from your audio interface.
 - b. Set volume as appropriate.
 - c. If using only one channel of input, check the **Mono** checkbox.
4. Adjust the **Output** channel volume to make it audible. You should now hear yourself. If using live mics, watch out for feedback.
5. In the **Remote 1** channel:
 - a. QuaxTrip uses the **Server** foo.ucsd.edu by default. This public “conniption” server helps make the connection but once connected the audio runs straight between the two computers.
 - b. Set the **Call Name** to something likely to be unique (e.g. jack-and-jill), no white space allowed. The **Call Name** is the same for you and your partner.
6. Set the **On/Off** toggle to on.

Once you and your partner have completed these steps, the connected checkbox should be checked and you should now be able to hear each other! If you’re experiencing dropouts, see below for how to fine-tune the settings.

If you want to play with more than one partner, use **QuaxTrip Quintet.maxpat** instead and repeat steps 5 & 6 for each partner, making sure to select a unique **Call Name** for each set of partners.

Advanced Settings

- The **Mute** toggle boxes mute the audio to your local **Output**. *They do not stop the audio from being sent to your remote partner(s).*
- The **Mono** toggle box in the Local channel sends the first (left) channel to both channels. In this case you should use only one channel to send to your remote partners, but the **Mono** toggle will make sure that the local input is sent to both left and right of your local output.
- The **Input Delay** can be set to absorb network time variations (in units of 64 samples). Maximum **Input Delay** is 100. The **Fill** counter shows how far ahead of the incoming stream you are. If this number drops below 2 there will be a dropout in the audio output. Increasing the **Input Delay** should decrease dropouts at the expense of latency.
- If your signal is mono, set **Channels** to 1 to save bandwidth. Note that this need not be symmetrical. That is, you could send 1 channel while your partner sends you 2.
- **Block Size** sets the size of the chunks of the audio you send. Increasing this number may decrease dropouts at the expense of latency.
- Check **2X** to send each outgoing packet twice (this doubles the data rate but might give fewer dropped packets).
- Be sure your sample rate (set in **Options -> Audio Status** in Max) matches your partner’s - if they don’t you’ll get breaks in the sound.

- If the **Calling Server** button is flashing you're either waiting for a response from the server, or from the partner. Once both of these arrive the **Connected** toggle turns on. At this point a call is in progress and the "packets in" counter should be changing. Press the **Reset** button to set **Packets In** and **Dropouts** counts back to zero.
- QuaxTrip automatically sets Max's **I/O Vector Size** and **Signal Vector Size** to 64 and engages **Scheduler in Overdrive**. Changes to these settings may result in distortion and crashes.
- To test latency:
 - a. Both you and a remote partner click **Latency Tester** in the appropriate **Remote** channel.
 - b. One partner selects **Measure** in the **Latency** window pop-up.
 - c. The other partner selects **Loop-back** in the **Latency** window pop-up.
 - d. The latency reading will show up in the partner that has selected **Measure**.
- Click the **Save Settings** button to save the current server, volume, etc. settings to the location of your choice on your hard drive. Click the **Load Settings** button to load a previous saved setup.
- QuaxTrip includes a rudimentary messaging system to allow the sending of text messages to the other participants. Click the **Messages** button and type a message in the pop-up window to send a text message to all connected peers. Text messages received appear in the Max Console.

Using the Connption Server

By default quacktrip uses a 'connption' server at foo.ucsd.edu, port 3840 (hexadecimal "f00") to connect calls. This should work for most people in most situations. You can, however, enter an alternative server name or IP address in the **Server** box if you wish. A connption server is built into QuaxTrip.

Computers on the Same Local Network

If you want to make calls between computers behind the same router (i.e. your local network), you will need to run a connption server behind that same router. Calls between two patches on the same computer won't work. On the computer you wish to be the server, click the **Connption Server** button to view your IP address. Set the **Server** of all computers to that listed as the **Local IP Address** in the **Connption Server** window. If you see more than one **Local IP Address** listed that means you have more than one local connection, probably both Ethernet and WiFi. Either turn off WiFi to avoid the ambiguity or check your computer's network settings to find out which IP address is currently active.

Over the Internet

If using your computer as a connption server with other computers over the internet, set the **Server** of all computers to that listed as the **Public IP Address** in the **Connption Server** window of the computer functioning as a server. Before this will work, you will need to [set up port forwarding](#) for port 3840 on the server computer (note that this step is only necessary when connecting computers over the internet, not for computers on the same local network).

Routing Audio to other Max Patches

QuaxTrip includes a few `send~` and `receive~` objects intended to allow you to send audio to and/or from your own Max patches. While QuaxTrip is open, open **Routing Audio.maxpat** in QuaxTrip's lib directory to see a demonstration of the various ins and outs available.

Sending Audio to the Local Channel

Using `qxt_local_patch_L` and `qxt_local_patch_R` `send~` objects in your patch you can send audio from your patch to the **Local** channel, bypassing the **Local** audio input. To set QuaxTrip to receive this audio:

1. Select send `qxt_local_patch` in the **Input Starting Channel** pull-down of the **Local** channel.

Using the Local Bus

Using `qxt_local_bus_send_L` and `qxt_local_bus_send_R` `receive~` objects plus `qxt_local_bus_receive_L` and `qxt_local_bus_receive_R` `send~` objects, you can route audio received from the **Local** input to your own patch and back again. To use this effects loop:

1. Select the proper channel for your audio interface in the **Input Starting Channel** pull-down in the **Local** channel.
2. Check the **Local Bus** checkbox in the **Local** channel.
3. Include `qxt_local_bus_send_L` and `qxt_local_bus_send_R` `receive~` objects in your patch to receive the audio.
4. Include `qxt_local_bus_receive_L` and `qxt_local_bus_receive_R` `send~` objects in your patch to send the audio back to QuaxTrip.

Using the Output Bus

Using `qxt_output_bus_send_L` and `qxt_output_bus_send_R` `receive~` objects plus `qxt_output_bus_receive_L` and `qxt_output_bus_receive_R` `send~` objects, you can route audio received by the **Main** output to your own patch and back again. To use this effects loop:

1. Check the **Output Bus** checkbox in the **Output** channel.
2. Include `qxt_output_bus_send_L` and `qxt_output_bus_send_R` `receive~` objects in your patch to receive the audio.
3. Include `qxt_output_bus_receive_L` and `qxt_output_bus_receive_R` `send~` objects in your patch to send the audio back to QuaxTrip.

Post-mute Audio

The audio that is being sent out to the **Output** of QuaxTrip is available via `send~` objects.

To set your patch to receive the post-mute **Output** audio:

1. Add **qxt_post_mute_L** and **qxt_post_mute_R** receive~ objects in your patch.

Pre-mute Audio

In addition to the audio sent to the **Output** of QuaxTrip, there are send~ objects that route audio *before* the **Mute** toggle. This allows you to mute channels locally for monitoring purposes while also sending all channels to another patch. For example, you might create a patch that receives the un-muted stream in order to livestream over the internet. The pre-mute sends are also pre-fader.

To set your patch to receive the pre-mute **Output** audio:

1. Add **qxt_pre_mute_L** and **qxt_pre_mute_R** receive~ objects in your patch.

To set your patch to receive the pre-mute audio from individual channels:

1. Add **0_qxt_pre_mute_L** and **0_qxt_pre_mute_R** receive~ objects in your patch to receive the **Local** channel.
2. Add **#_qxt_pre_mute_L** and **#_qxt_pre_mute_R** for each **Remote** channel (e.g. **1_qxt_pre_mute_L** and **1_qxt_pre_mute_R** for **Remote 1**).

Using the Messaging System

In addition to the audio that is sent between peers, there is also a system for sending messages. Messages can be symbols, numbers, or lists.

To send messages:

1. Add a **qxt_send_message** send object in your patch.
2. Send the message **n the message** where **n** is the Remote channel number where you wish to send the message and **the message** is the symbol, number, or list you wish to send.

To receive messages:

1. Add a **qxt_receive_message** receive object in your patch.
2. Messages received will be in the format **n the message** where **n** is the Remote channel number that sent the message and **the message** is the symbol, number, or list sent.

For an example, see the patch **lib/qxt.Messages.maxpat**. This is the patch that implements the rudimentary texting system of QuaxTrip.

QuaxTrip Copyright 2020 Damon Holzborn. QuaxTrip is open source software, which you can use and modify under the MIT License (LICENSE.txt in this distribution).

Also included in this distribution is Quacktrip. Quacktrip is Copyright 2020 Miller Puckette. Quacktrip is open source software. free to use and modify under the Standard Improved BSD License (quacktrip/lib/LICENSE.txt in this distribution).