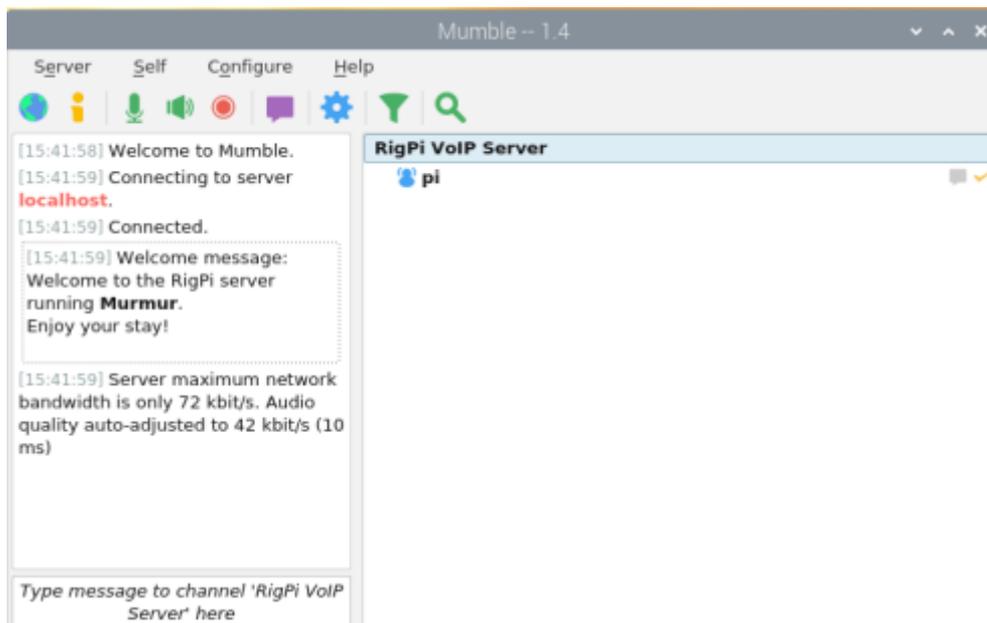


Mumble

Mumble is an open source VoIP system. A VoIP server called Murmur is installed on RSS along with the Mumble client. The RigPi Audio board normally provides an input and output path for Mumble client, although USB audio sound devices or radio CODECs can also be used. One radio can be connected to the Mumble client. Remote Mumble clients can also connect to the server and receive and send audio. Mumble clients are available for all popular operating systems.

The Murmur Server is named RigPi VoIP Server and the configured Mumble Client is named pi. Mumble can be started manually from the RSS Desktop Menu by selecting Internet>Mumble. A utility named Mumble Autostart is in the Internet menu as well as on the RSS Desktop. Run Mumble Autostart to have Mumble start automatically when RSS is rebooted.

When you start Mumble the Mumble window opens on RSS:



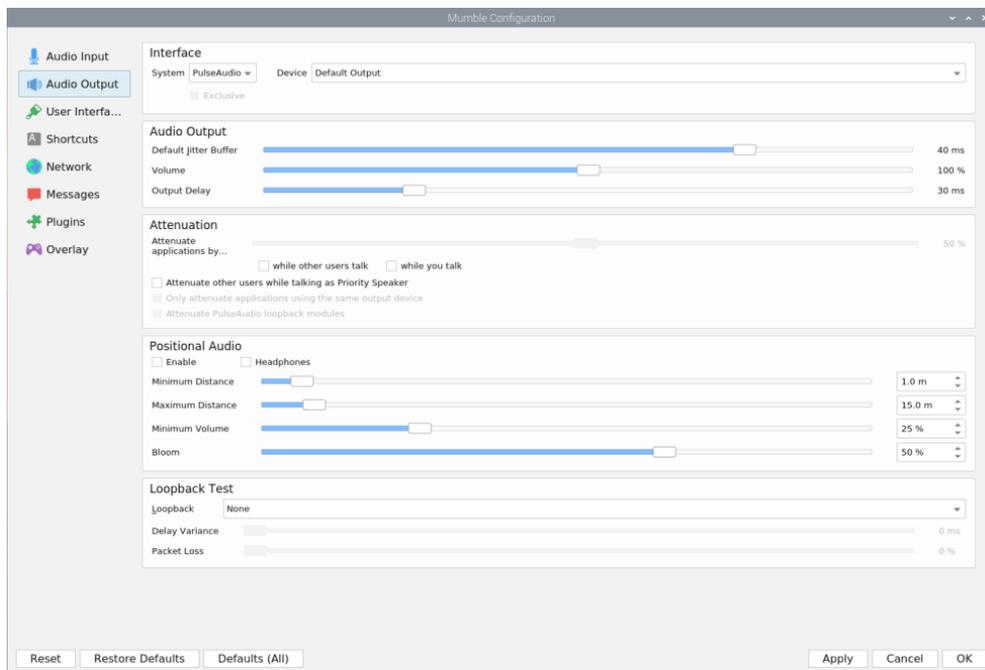
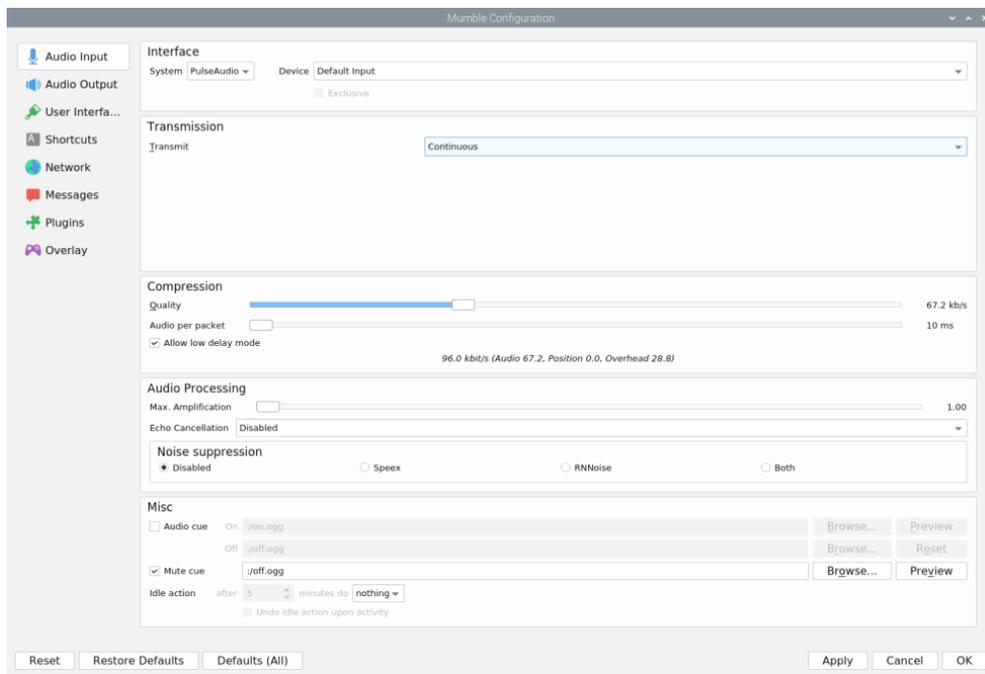
Mumble Settings

If you make any changes to Mumble settings, click Apply at the bottom of the settings window to test. When you are satisfied, click OK. Restart Mumble to have the settings stick.

Mumble uses RigPi Audio for input and output audio, audio USB dongles, or CODEC's from your radio. Set Device to Default Input in Audio Input and Default Output Audio Output.

Refer to Mumble Help for setting up Mumble. One setting that applies to Mumble in RSS is found in the Audio Processing group of Mumble Configure>Settings>Audio Input. Set the Audio Processing to Disabled. If Noise Suppression is enabled the normal background noise on HF can cause distortion and clicking in the audio output. (See screen shot below.)

Mumble can send audio continuously, by use of a PTT key, or by detecting the presence of audio. The screen shots below are shown below to include all audio in and audio out settings. The settings shown are typical. You will have to adjust them to maximize audio quality with your RigPi.



In Mumble Configure>Settings>Network enable the 'Reconnect automatically' and 'Reconnect to last server on startup' options.

Audio Input

Transmit: This lets you change what causes audio from your radio to be sent. I recommend "Continuous", but you can also have Mumble attempt to transmit only when there is active audio from your remote device.

Audio Cue: This causes a sound to play whenever you press or release the Push-To-Talk key. Not active when Continuous Transmission is selected.

Idle action: This will cause Mumble to automatically mute the receive audio if there is no radio audio for a certain amount of time. Dragging this slider all the way to the left will disable this feature.

Quality: This is the quality at which Mumble will encode audio from your radio. I recommend that you leave this at **40.0 KB/s**. However, if the audio sounds distorted, you can try raising or lowering this value.

Audio per packet: This is the amount of audio that Mumble sends at a time. A lower value means less latency, but might potentially introduce quality loss, if too low. I recommend that you leave this at **20 ms**.

Noise Suppression: Mumble can automatically filter out static background noise. When set to Speex, RNNoise or Both, a slider will adjust how aggressively noise is filtered. If filtering is too aggressive, then your voice may become distorted. I recommend setting Noise Suppression Disabled to remove all noise suppression.

Max. Amplification: This setting controls how loud the receive audio is.

Audio Output

Default Jitter Buffer: If your mic audio is distorted, metallic, or sounds like a robot, try increasing this setting a notch or two.

Volume: Output volume. The volume at which audio is sent to the mic/audio input.

Output Delay: If your transmit audio is distorted or is breaking up, try increasing this value.

Attenuate applications by...: This setting determines how much Mumble will reduce other applications' volume when people talk. A setting of 100% means to mute other applications; 0% means to not do anything to other applications. Additionally, you can select to attenuate when others talk, and/or when you talk with the "while other users talk" and "when you talk" check boxes.

Loop back: This is a super handy feature for troubleshooting problems with your transmit audio settings. You can set Mumble to "loop back" to you, so you hear what you are saying the way other people will hear you. When troubleshooting, try setting this to "Server", and then tweak options in Audio Input and Audio Output to get your audio sounding just right. Remember to set it back to "None" after you're finished, or else nobody will be able to hear you

Managing Accounts

If you wish to remove clients that have been registered, set up a new user with username SuperUser. The SuperUser can delete registered accounts.

Using a Remote Server

RSS is can use an external Mumur server. If RSS is connected to the Internet and you expect to have a large number of users, a remote Murmur server may be attractive.

Murmur servers are available for free or low cost. Commercial server companies typically charge based on the number of "slots" you want to use. A slot is a unique user logged in to the server. Server fees typically run less that \$2 per month per slot. Details for connecting to a remote server are beyond the scope of this manual, but instructions are usually given on the server you choose.

Running more than one Mumble client

Since RSS is multi-user, one wonders if it is possible to run audio from more than one radio through RSS. The short answer is yes! RSS can serve multiple users for rig control AND audio.

There are two requirements.

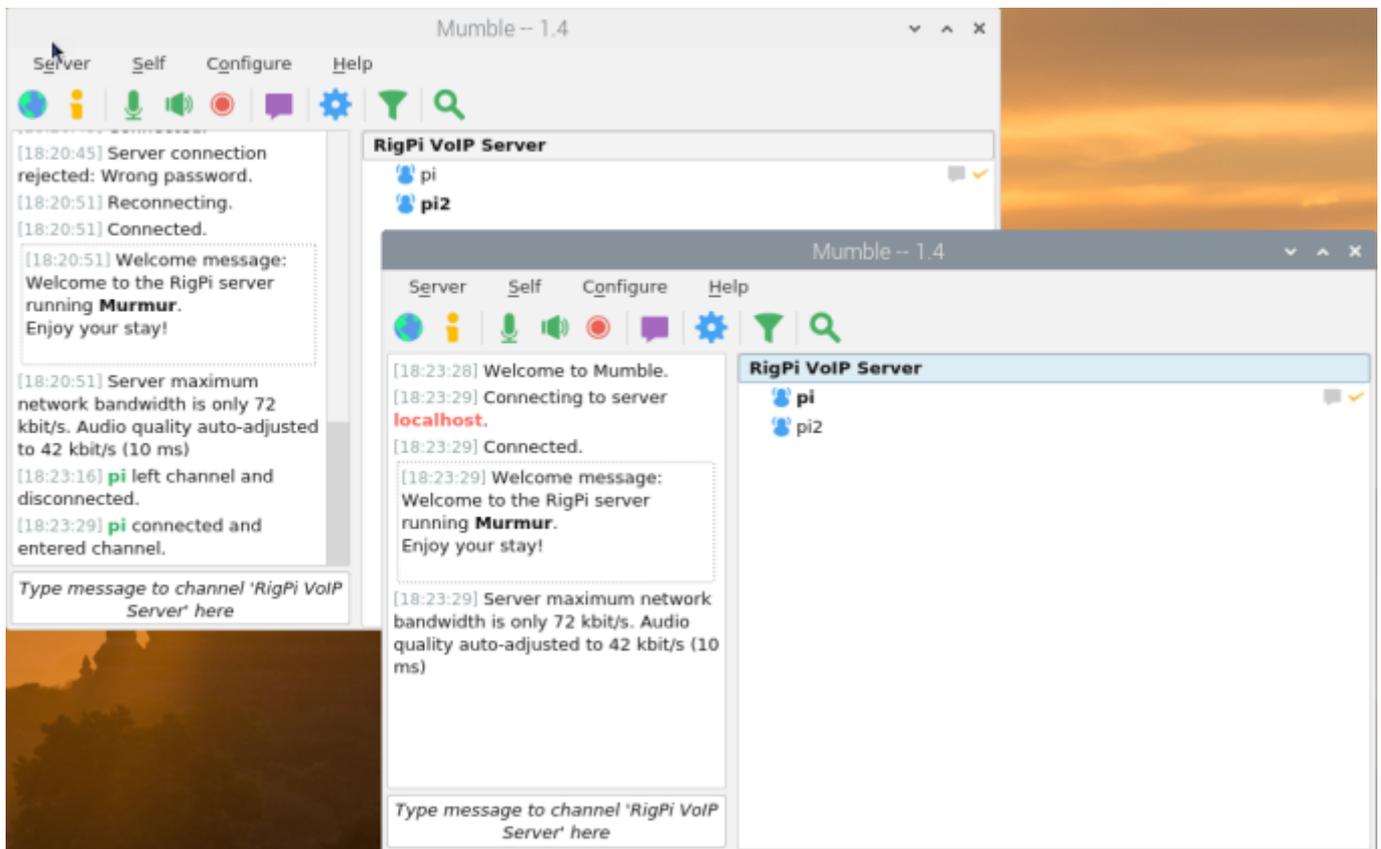
1. You need one (or more) audio IO devices in addition to RigPi Audio. This can be done with a USB audio dongle. For my tests I used a Griffin dongle, but any inexpensive USB/Audio device should work well.
2. The second (and subsequent) Mumble clients must be started in a special way.

Close any open Mumble clients on the Raspberry Pi desktop. To start the first instance, go to the RSS Desktop Menu and open Run... In the Run box, type

```
/home/pi/mumble/build/mumble -m -n
```

then press enter. The -m parameter signifies a multiple instance. The -n parameter tells Mumble to ignore security certificates.

Create a new Servername in the Mumble Server Connect dialog, such as "Radio2". Edit the connection and use a different Username, such as pi2. Enter localhost for the Address. Click OK, then connect. You will see two instances of Mumble on your desktop.



Now you can log in. I used pi2 as the username. With two clients you see both users in the client lists, pi and pi2. The user connected with its primary client is shown in bold. Assign the audio dongle to the second client. Now you can connect the audio to/from two rigs.

Since all users hear all other users, you will hear audio from both radios. That has interesting possibilities, such as monitoring 2 repeaters at once. You may want to keep the audio from the two radios separate, in which case there is a way to do that. Mumble provides channels, users can be assigned to channels, and the audio is separated. From a remote device you can create two accounts, one for each channel. To

switch from one channel to the other, log out from the Mumble client on the remote device and log in to the other account.

To create channels, log in to Mumble with the username SuperUser. Superusers can create channels and assign users to default channels. Here are more details:

<https://support.mayfirst.org/wiki/mumble-private-channel>

Install Bonjour

The Mumble server requires that Bonjour be installed on your computer for clients to connect. If you are using a Windows computer, follow the instructions [here](#). Mac computers already have Bonjour installed.