

# BUTT (1.44.0) Manual

Daniel Nöthen

# Table of Contents

1. About .....	1
2. Install .....	1
3. butt-client .....	2
4. Quick start .....	2
5. Install AAC libraries .....	2
6. Configuration .....	3
7. Main Window .....	3
8. Audio Mixer .....	4
9. Audio .....	4
10. Streaming .....	5
11. Shoutcast .....	7
12. Icecast .....	7
13. Secure Connection over SSL/TLS (Icecast only) .....	7
14. WebRTC (WHIP) .....	8
15. Recording .....	8
16. DSP .....	10
17. MIDI .....	11
18. GUI settings .....	13
19. The BUTT agent (Windows only) .....	14
20. Command line control .....	14
21. Command line options .....	14
22. Uninstall .....	17
23. Contact .....	17
24. Donate .....	17

# Chapter 1. About

BUTT (Broadcast Using This Tool) is an easy to use, multi OS streaming tool.

It supports Icecast, Shoutcast and WebRTC and runs on Windows, macOS and Linux.

The main purpose of BUTT is to stream live audio data from your computers microphone or line input

to an Icecast, Shoutcast or WebRTC (WHIP) server. If you want you can also record your broadcast locally

on your computer.

With BUTT you can not directly stream audio files. However, with the help of tools like

Voicemeeter (Windows), Blackhole (macOS) and PulseAudio (Linux) you can also stream audio files directly from your favorite music player.

# Chapter 2. Install

*OS X:*

Mount the **butt-<version>.dmg** by double clicking and drop the butt file on the Applications folder.

*Windows:*

Just run the **butt-<version>-setup.exe** and follow the installer pages.

The default installation path might be unusual, but this way it is possible to install BUTT without administration rights.

*Linux/MinGW (Windows):*

Note: If you want to only install the butt-client, skip directly to the butt-client section below.

First of all the following libraries have to be installed on your system

*ftk-1.3, portaudio19, libmp3lame, libvorbis, libogg, libflac,*

*libopus, libsamplerate, libfdk-aac,*

*libdbus-1, 'libcurl, libssl', 'portmidi*

They are quite common and should be included in every popular linux distribution.

On **Ubuntu** you can install them with

```
sudo apt-get install libfltk1.3-dev portaudio19-dev libopus-dev libmp3lame-dev libvorbis-dev libogg-dev libflac-dev libfdk-aac-dev libdbus-1-dev libsamplerate0-dev libssl-dev libcurl4-openssl-dev libportmidi-dev
```

On **openSUSE** you can install them with

```
sudo zypper in fltk-devel portaudio-devel libmp3lame-devel libvorbis-devel libogg-devel flac-devel libfdk-aac-devel libopus-devel libopencore-amrnb-devel libopus-devel libsamplerate-devel dbus-1-devel libcurl-devel portmidi-devel
```

On Distributions which don't have libfdk-aac like **debian** you can compile without aac support with:

```
./configure --disable-aac
```

For compiling on **Windows** I recommend the msys2 x64 ([www.msys2.org](http://www.msys2.org)) environment.

They have all the needed packages in their pacman repository. Additionally you need to install the

`libwinpthread-git` package.

Make sure that you select the x64 version of each package.

After installing the above libraries you can install BUTT from source as usual:

- `tar -xzf butt-<version>.tar.gz`
- `cd butt-<version>`
- `autoreconf -i` (only on Windows/MSYS2)
- `./configure`
- `make`
- `sudo make install`

In case the included configure script or the make process fails on your system, try to create a new configure script by invoking: `autoreconf -i` and start with `./configure` again.

## Chapter 3. butt-client

`butt-client` is a minimal binary which can be used to send commands to a running BUTT instance.

There are two ways to build `butt-client`:

1. Configure the build by adding `--with-client` to the configure command.
2. Configure it by adding `--without-butt`.

The first option builds both BUTT and `butt-client` and it requires all system dependencies.

The second option builds `butt-client` only and it requires none of the system dependencies.

This is the recommended option if you only need `butt-client`, especially if you are installing it on a different machine. This is because the resulting binary is much smaller and it does not link with any system dependencies, except the standard ones, which means that it is portable as long as you copy it to another machine which shares the same operating system as the one you are building it on.

## Chapter 4. Quick start

When you start BUTT the first time, it will create a default configuration file in in your home directory (`~/buttrc`) on Linux and macOS or at `C:\Users\<username>\AppData\Roaming\buttrc` on Windows.

In order to connect to a server, you need to add a new server in the config window.

Just open the settings window and click on [ADD].

Now fill in the input fields with the server credentials and click on the [ADD] at the lower right corner.

To connect to the server just press the play button in the main window and enjoy your broadcast.

## Chapter 5. Install AAC libraries

*Windows:*

1. Download libfdk-aac-2.dll from [here](#)
2. Go to the installation directory of BUTT by typing %LOCALAPPDATA%\butt into the file explorer
3. Copy libfdk-aac-2.dll into the installation directory of BUTT
4. Restart BUTT
5. Enjoy AAC streaming

*MacOS:*

1. Download and install butt\_libfdkaac\_macOS.pkg from [here](#)
2. Restart BUTT
3. Enjoy AAC streaming

Note: The installer will install libfdk-aac.2.dylib into /Library/Application Support/butt/

*Linux:*

In case your Linux distribution ships BUTT without AAC support, you need to install BUTT from source. Please refer to the [Install](#) section for more information on that.

## Chapter 6. Configuration

The command line option `-c <path_to_file>` allows you to define a new standard configuration path.

This makes it possible to have multiple instances with different configurations running. In case the file does not exist, BUTT will create a default file for you.

A tutorial on how to use the `-c` command line option to run multiple BUTT instances can be found [here](#).

[Save]: Saves your current settings to the standard configuration file or to the file that was passed to the `-c` option

[Export]: Saves your current settings to the given file

[Import]: Loads the selected file and applies the settings

### CAUTION

If you use the `-c` command line option and import another configuration file by using the import function, pressing [Save] will overwrite the file that was passed to the `-c` option.

In some cases the configuration file can become damaged and BUTT will not be able to start correctly. In that case you can hold down the shift key during BUTT startup to create a new configuration file.

## Chapter 7. Main Window

The dot matrix display shows you the current state of the BUTT software.

The states are: idle, streaming, recording.

When in streaming and/or recording state you can cycle through the information by clicking on the

display.

You can choose between online duration, data sent, recording duration, data recorded and On Air state

The [>] symbol shines yellow if BUTT is connected to a server.

The [O] symbol shines orange if the [start rec. when connected] checkbox is activated.

The [O] symbol shines red if BUTT is currently recording.

Additionally the play and record buttons change their color to yellow and red when BUTT is connected or recording.

Gain slider: The slider is only visible when the log window is visible as well.

With this slider you can attenuate and amplify the input signal between  $-24\text{ dB}$  and  $+24\text{ dB}$ , respectively.

Double clicking the slider handle resets the gain to  $0\text{ dB}$ . Use this slider only to fine tune your input signal.

To enter the gain as a number with your keyboard, right click the slider handle. The slider does not change the operating systems input volume setting. Instead, the input signal is multiplied by the given factor. Thus, adding too much gain will also add lots of noise.

VU meter: The vu meter shows the current streaming or recording volume in dBFS.

## Chapter 8. Audio Mixer

Click on [Mixer] to open the Audio Mixer. The mixer allows you to individually adjust the volume of your audio devices and have different volumes for streaming and recording. The Master Gain slider is coupled with the gain slider on the main window.

The crossfader allows you to blend volumes between the primary and secondary audio device. Moving it from the center to one side lowers the volume of the other side.

## Chapter 9. Audio

The audio settings tab allows you to select your primary and secondary audio interface and the desired sample rate.

If you have a multi-channel audio device you may also select the desired input channels for the left and right audio channel.

If you want to stream in stereo, make sure that the left and right channels are mapped to different input channels like 1 and 2. Otherwise you will only stream a mono signal.

With the channel mode setting you tell BUTT if the audio stream shall be encoded to stereo or mono.

This brings us to 4 possible combinations.

*Channel mode = Stereo and different input channels are selected for Left and Right:*

This is probably the most used combination.

Left and Right channels are encoded into a stereo stream.

*Channel mode = **Stereo** and the **same input channel** is selected for Left and Right:*

The selected input channel is used for the left and right channel in a stereo stream.

*Channel mode = **Mono** and **different input channels** are selected for Left and Right:*

Left and Right channels are averaged into a mono stream.

*Channel mode = **Mono** and the **same channel** is selected for Left and Right:*

The selected channel is used as mono channel in a mono stream.

The "Remember Device by" setting lets you choose if BUTT remembers the current device by its ID or its name. As the ID may change when changing the number of connected audio devices on your computer

or when updating your operating system, it is recommended to let BUTT remember the audio device by its name.

Clicking on [**Update devices**] will rescan the system for audio devices and update the Primary and Secondary audio device list accordingly.

Within the Streaming and Recording section you can select individual audio codecs and corresponding bitrate for streaming and recording. Advanced users have the ability to change codec settings by clicking the gear symbol.

The Signal detection levels define at which level BUTT should treat the signal as present or absent. These levels are used by the auto connect/disconnect or recording feature described below.

## Chapter 10. Streaming

Before you can start streaming you need to add a server first. If you have not added a server yet, please refer to the quick start section of this manual.

To start streaming just click the play symbol.

BUTT will try to connect to the server until you press the stop button.

If the connection gets lost, BUTT will try to reconnect until the stop button is pressed

You can stream in 5 different audio codecs: mp3, aac+, ogg/vorbis, ogg/opus and ogg/FLAC.

In case opus is selected the sample rate is always resampled to 48 kHz.

If you want to prevent BUTT from resampling, you should select 48 kHz as the device sample rate.

*Song name:*

If you want to inform the listener about which song is currently playing you can do that on the [**Stream**] tab.

You only need to type the song into the **Song Name** input field and hit Enter or click [**OK**].

BUTT can also update the song automatically from a text file. The first or the last line of the file must be the name of the song. As soon as BUTT detects that the file has been changed, it updates the name of the song on the server.

To retrieve the current song name from an URL you can enter the URL into the *Update song name from URL* field and enable the *Activate* checkbox. Set the *Update interval* to the number of seconds you want BUTT to check for a new song name. BUTT uses the first line from the server response for the song name update. HTML tags are not parsed nor removed, so it is recommended use a URL that returns a simple text file instead of a HTML formatted website.

If you run BUTT on macOS or Linux you can even transfer the current song name from an audio player to BUTT.

Supported audio players:

Linux: Rhythmbox, Banshee, Clementine, Cantana, Spotify

MacOS: iTunes/Music, Vox, Spotify

As broadcasting with Icecast and Shoutcast is not realtime, the listener receives the audio content with a few seconds of delay. This delay introduces an offset between the current song name and the actual song that is playing. To prevent confusion on the listeners side you can add a delay to the automatic song update.

In case you want to add a prefix and/or a suffix to your song name you can do that by entering the desired text into the corresponding input field.

*Stream infos:*

In the **[Main]** settings window you can add stream infos. This allows you to deliver more details about your stream. For example the genre of your music, description of your station, web address etc.

Unfortunately, it is not possible to update stream infos during a broadcast. You need to reconnect for updating the stream infos.

For the name and description fields you may use the same placeholder variables as you can in the recording file/path. Activate the *Expand variables* checkbox to enable it.

*Automatic streaming:*

If you activate the checkbox *Start streaming after launch* BUTT will automatically connect to the server as soon as the application has been started.

BUTT can also connect and disconnect depending on the audio signal level:

To connect automatically if a signal is present for a certain amount of time enter an integral number larger than 0 into the *Start if signal is present for [...] seconds* field and activate the checkbox.

To disconnect automatically if the signal is absent for a certain amount of time



enter an integral number larger than 0 into the *Stop if signal is absent for [...] seconds* field and activate the checkbox.

The default signal detection levels are set to -50.0 dB and can be independently changed for the present and absent signal cases in the [\[Audio\]](#) tab.

## Chapter 11. Shoutcast

Basically BUTT is a Shoutcast 1 source client and does not support the native Shoutcast 2 protocol. However, Shoutcast 2 servers are compatible with Shoutcast 1 client by default. So you can also connect to Shoutcast 2 servers with BUTT.

To add a Shoutcast server, open the settings and click on [\[ADD\]](#). Now enter the server address, port and password. If you did not setup the Shoutcast server by your own, you will get the required information from your streaming hoster.

If you need to enter Shoutcast 2 specific login data like a user name or a stream ID you can add this to your password as follows:

Let's assume your user name is **MyUser**, your password is **MyPassword** and the **stream ID** is **3**. To add the user name to the password field enter the following:

```
MyUser:MyPassword
```

The stream ID 3 can be appended after the password like this:

```
MyUser:MyPassword:#3
```

## Chapter 12. Icecast

To add an Icecast server open the settings and click on [\[ADD\]](#). Now enter the server address, port and credentials like, user, password and mounpoint. If you did not setup the Icecast server by your own, you will get the required information from your streaming hoster.

## Chapter 13. Secure Connection over SSL/TLS (Icecast only)

To enable encryption for an icecast server, you only have to activate the *Use SSL/TLS* checkbox in the server settings. Please bear in mind that the server must be configured with SSL/TLS support in order to make this working. The connection will fail if you activate SSL/TLS for a server which does not support encryption.

If the certificate validation fails, BUTT will ask you if you want to trust

that certificate anyway. If you click on **[TRUST]** BUTT will establish the connection and remembers the decision for that certificate and server. By pressing the button *[Revoke certificate trust]* you can revoke that decision.

If you want to specify your own file or folder with CA certificates, you can enter the path to the file or folder in the **[TLS]** tab of the settings window. Usually you should not need to enter any information there.

## Chapter 14. WebRTC (WHIP)

Since version 1.42.0 BUTT is able to stream to a WebRTC WHIP server.

WebRTC is a relatively new technology for realtime peer-to-peer communication for video and audio. Although WebRTC is point-to-point by default, with the additional WHIP protocol it is also possible to have a client/server connection like it is used by Icecast and Shoutcast.

The biggest advantage of using WebRTC is its low latency which can be lower than 200 ms. However,

to accomplish such a low latency, the WebRTC protocol has to make use of certain techniques that require a perfectly stable connection to the server. Otherwise small interruptions of the audio signal

may occur or the audio signal may sound speed up or slowed down for a short period of time.

Because of this, it is not recommended to use WebRTC for streaming music but rather use it for situations where you need near real time voice broadcasting like for a live commentary of a sports event.

To add a WebRTC server open the settings and click on **[ADD]**. Now select WebRTC and enter your WHIP URL.

If needed you can also enter a STUN or TURN server address into the ICE server field and a Bearer Authentication Token.

WebRTC does only work with opus, so please make sure that you have opus selected als streaming codec.

## Chapter 15. Recording

BUTT is able to record and stream simultaneously in different bit rates.

For example you can stream with 96 kbit and record with 192 kbit.

Recording is possible in mp3, aac+, ogg/vorbis, ogg/opus, FLAC or wav.

To record your session you first need to select the destination folder and specify a file name in the **[Record]** tab.

BUTT will replace specific date variables with the current time and date. For example `rec_(%m_%d_%y).mp3` expands to `rec_(03_28_2008).mp3`.

Other possible time variables are **%H** (hours) **%M** (minutes) **%S** (seconds). Refer to the table below for more supported date variables.

With the **%i** variable you can add an index number to your file name.

This means with `rec_%i.mp3` BUTT first tries to record to `rec_0.mp3`. In case that

file already exists, BUTT tries `rec_1.mp3` ...

To manually start the recording simply press the record symbol.

To stop recording just click on the record symbol again.

#### *Automatic recording:*

If the *start recording when connected* checkbox is activated BUTT starts the recording immediately after a connection with the server has been established.

Vice versa BUTT will stop the recording if the *Stop recording when disconnected* checkbox is active.

Additionally you can tell BUTT to immediately start recording after the application has been launched by checking the *Start recording after launch* box.

To start recording automatically if a signal is present for a certain amount of time enter an integral number larger than 0 into the *Start if signal is present for [...] seconds* field.

To stop recording automatically if the signal is absent for a certain amount of time enter an integral number larger than 0 into the *Stop if signal is absent for [...] seconds* field.

The default signal detection levels are set to -50.0 dB and can be independently changed for the present and absent signal cases in the [\[Audio\]](#) tab.

You can also tell BUTT to split your recording into separate files every **n** minutes. Just enter a number higher than 0 into the *Split file every [...] minutes* field.

Let's assume your file name is `rec_(%m_%d_%y)_%i.mp3` Then the first file is expanded to `rec_(03_28_2008)_0-1.mp3`, the second after **n** minutes to `rec_(03_28_2008)_0-2.mp3`, the third to `rec_(03_28_2008)_0-3.mp3`, you got it.

If the *sync to full hour* checkbox is activated the automatic file splitting is synchronized to the full hour. This means if the current time is 8:55 and file splitting is set to *15 minutes*, the second file starts at 9:00 and the third at 9:15.

If you want to split the recording now, just click the *[Split now]* button.

#### **Supported variables:**

Variable	Meaning
%N	Server name
%a	abbreviated weekday name (e.g. Fri)
%A	full weekday name (e.g. Friday)
%b	abbreviated month name (e.g. Oct)
%B	full month name (e.g. October)
%d	day of the month, as a number (1-31)
%H	hour, 24 hour format (0-23)

Variable	Meaning
%I	hour, 12 hour format (1-12)
%j	day of the year, as a number (1-366)
%m	month as a number (1-12).
%M	minute as a number (0-59)
%p	AM or PM
%S	second as a number (0-59)
%U	week of the year, (0-53), where week 1 has the first Sunday
%w	weekday as a decimal (0-6), where Sunday is 0
%W	week of the year, (0-53), where week 1 has the first Monday
%y	year in decimal, without the century (0-99)
%Y	year in decimal, with the century
%Z	time zone name

## Chapter 16. DSP

The DSP section contains two sound processors. A 10-band EQ and a Dynamic Range Compressor. With the *Stream* and the *Rec* checkboxes you can decide if a sound processor should be activated for the streaming and/or recording signal chain.

### *10-Band Equalizer:*

The equalizer allows you to change the gain of certain frequency bands from -15 dB to 15 dB. Note for professionals: The equalizer uses a Q of 2 on each frequency band.

### *Dynamic Range Compressor:*

Dynamic range compression is used to reduce the difference between loud and quiet parts of the signal, and thus provide a more consistent experience for listeners. It is used by virtually all professional radio stations.

The recommended procedure for configuring the compressor is as follows:

1. Start playing the loudest audio source you intend to broadcast (typically music), and line it up with the master gain slider.
2. Enable the compressor, and adjust the threshold and gain to suit. The attack and release times can generally be kept as they are, unless you have a particular reason to change them. You will notice that the overall signal level goes down, as it is being compressed.
3. Adjust the makeup gain to bring the signal back to its original level.
4. Now test with a quieter audio source (such as your voice), and see that the level of that is boosted in comparison. If the quieter source

is still too quiet, reset the makeup gain to 0 and repeat from step 2 onwards, until you have a satisfactory result.

This procedure can take some time to find the optimum settings, which are determined by listening as much as by metering, but it generally only needs to be done once - BUTT will save your settings, so once you have values that work well for your content, you probably don't need to adjust them again.

As a rough guide, music should be compressed relatively subtly, with a fairly high threshold and a ratio typically between 2 and 3. Pure speech content can be compressed much more dramatically, with a low threshold and a ratio of 5 or more; this will make the speech easier for the listener to understand, and will also reduce the differences between different speakers or by not keeping a very consistent distance from the microphone.

For mixed speech and music broadcasting, it is recommended to set BUTT's compression as for music, and then have an additional compressor (typically a hardware module) between the microphones and the final mix.

To check if the signal power exceeds the threshold, the compressor usually averages the signal power over time and compares it with the threshold. This averaging process reduces the speed of the compressor. In case you need a very fast responding compressor you can activate the *Aggressive Mode* option. With this option enabled the compressor does not average the signal power over time anymore.

## Chapter 17. MIDI

In the [MIDI] tab you can set up a MIDI device to control BUTT via MIDI CC messages. To activate the MIDI interface simply select your MIDI device from the *MIDI Device* drop down menu.

The *MIDI indicator light* will blink if BUTT receives a MIDI message. This allows you to verify the connection between BUTT and your MIDI device.

The *MIDI Command* list shows you which parts of BUTT can be controlled via CC messages. By default all commands are disabled. To enable a command select the command from the list and click on **Enable**.

### MIDI implementation table

Command	CC value	Action
Start/Stop broadcasting	> 63	Starts broadcasting
	< 64	Stops broadcasting

Command	CC value	Action
Start/Stop recording	> 63 < 64	Starts recording Stops recording
Master Gain	0-127 (absolute mode)	Scales linearly between -24 dB and +24 dB
Streaming Gain	0-127 (absolute mode)	Scales linearly between -18 dB and +18 dB
Recording Gain	0-127 (absolute mode)	Scales linearly between -18 dB and +18 dB
Primary Device Gain	0-127 (absolute mode)	Scales linearly between -18 dB and +18 dB
Secondary Device Gain	0-127 (absolute mode)	Scales linearly between -18 dB and +18 dB
Cross fader	0-127 (absolute mode)	Blends between Primary (0) and Secondary (127) Device
Mute/Unmute Primary Device	Any	Toggles mute state
Mute/Unmute Secondary Device	Any	Toggles mute state

There are several options available to adapt BUTT to your MIDI controller:

*Channel:*

Select the channel where your MIDI controller sends MIDI messages to.

*CC:*

Type in a number from 0 to 127 which corresponds to the control element (button, knob, slider) of your MIDI controller.

*Learn:*

If you don't know the channel of your MIDI device or the CC number of your control element you can activate the **learning mode**. After pressing the *[Learn]* button, BUTT will wait for an incoming CC message and set the channel and CC number for you automatically.

*CC Mode:*

Volume controls can be controlled in two different ways:

1. In **absolute mode** BUTT maps the incoming CC values 0-127 linearly to its internal range. In case of a volume slider this means 0 maps to min and 127 to max volume.

If you switch back and forth between controlling a slider via the mouse and via your MIDI controller

the UI and your MIDI controller will get out of sync which will cause value jumps when controlling a slider via MIDI after you changed the slider position via mouse before.

To prevent those jumps activate the **Soft takeover** option.

2. The **relative modes** interpret the incoming CC values as increments or decrements.

Select the relative mode that fits to your MIDI controller.

## Chapter 18. GUI settings

Within the [GUI] settings you can change certain settings to your personal preferences.

### *Language:*

In case BUTT does not correctly detect your system language or you prefer a different one you can select your preferred language here.

After changing the language a restart of BUTT is required.

### *Display Color:*

Change the text and background color of the main display to your favorite color.

### *VU meter:*

Select between gradient or solid colors for VU meter. For better visibility the solid colors mode is recommended.

If the *Always show tabs* checkbox is unchecked the Streaming and Recording tabs on the VU meter are only visible if you hover your mouse over the meter.

### *Attach settings window to main window:*

If checked, the settings window will stick to the right side of the BUTT window.

### *Stay always on top:*

If checked, BUTT will always be on top of other application windows.

### *Remember main window position:*

If checked, BUTT will open on the previous position and screen.

In case the previous screen is not available BUTT will open on the default screen.

Additionally, this function also remembers the size of the log window.

### *Hide log window after start up:*

If checked, BUTT will open with the log window hidden.

### *Change display mode every 5 seconds:*

If checked, the display rotates automatically through the states online duration, data sent, recording duration and data recorded.

### *Start minimized:*

If checked, BUTT will start minimized to your Taskbar or Dock (macOS).

### *Disable gain control:*

If checked, the gain control slider on the main window will be disabled.

This prevents users from accidentally changing the volume.

### *Show listeners:*

If checked, BUTT retrieves the number of current listeners from your server every 3 seconds.

If available the number of listeners are shown on the lower right corner at the display.

**NOTE**

This feature only works with plain Icecast and Shoutcast servers. It will probably not work if you use any streaming service provider.

## Chapter 19. The BUTT agent (Windows only)

Version 0.1.29 has introduced the little helper tool called BUTT agent.

The main purpose of the BUTT agent is to make it possible to minimize BUTT into the windows system tray.

Minimizing to tray is not the only feature, though. You also have the ability to let it display balloon notifications for connects/disconnects and for song updates. If you want to start BUTT at windows start you can also enable/disable this with the agent.

BUTT starts the agent automatically if the checkbox *Minimize to tray* is enabled. If you want to use the balloon notifications you should activate the *Start agent at startup* checkbox in the [Main] settings tab.

Once the agent has been started you will find it in the system tray.

From here you can manage the minimized BUTT instances and activate the features mentioned above. A left click on the tray icon minimizes or raises the last BUTT instance. A right click opens up a context menu with more options.

The BUTT agent closes itself if no more BUTT instance is running.

## Chapter 20. Command line control

BUTT can be controlled from command line.

If you want you can even control BUTT from a remote computer.

Please refer to the section below for more information on that.

## Chapter 21. Command line options

BUTT has several command line options which can be separated into two modes which are the operating mode and control mode.

*Operating Mode:*

These options change the behaviour of the instance you are about to start.

*-c <path>:*

This option allows you to select a different configuration path. It is useful if you want to run several BUTT instances with different configurations. Just pass a different configuration file with the -c option for every instance.

*-L:*



Prints available audio devices.

**-A:**

This option tells BUTT to accept control commands from your network or even the internet.

**CAUTION**

When using this option everyone in your network or even internet may control your BUTT instance. Please use this option only if you have secured your network appropriately.

**-U:**

(uppercase U)

The command server will use UDP instead of TCP. If you pass the -U flag here you also need to pass it when sending control commands.

**-x:**

Use this option if you do not want to run a command server at all. This will also disable receiving commands from your local machine.

**-p <port>:**

With this option you can define the port of the command server. The default port is 1256. Use this option for example if you have several BUTT instances that you want to control from command line.

*Control Mode:*

With these options you can send control commands to a running BUTT instance. Note that you can send these commands from either BUTT or butt-client binaries.

**-s [name]:**

(lowercase s)

This command tells BUTT to connect to the server *name*. If the *name* parameter is omitted, BUTT will connect to the currently selected server.

**-d:**

When receiving this command, BUTT will disconnect from the current server.

**-r:**

This command starts the recording engine.

**-t:**

Use this option to stop the recording.

**-n:**

In case a recording session is active this option splits the current file like it would if you press the [Split now] button in the user interface.

**-q:**

Closes BUTT gracefully.

**-u <song name>:**

(lowercase u)

Sends a new song name to the server.

*-M <threshold>*:

Sends a new streaming signal threshold time to the server.

Minimum value is 0.1. Send 0 to disable automatic stream start.

*-m <threshold>*:

Sends a new streaming silence threshold time to the server.

Minimum value is 0.1. Send 0 to disable automatic stream stop.

*-O <threshold>*:

Sends a new recording signal threshold time to the server.

Minimum value is 0.1. Send 0 to disable automatic recording start.

*-o <threshold>*:

Sends a new recording silence threshold time to the server.

Minimum value is 0.1. Send 0 to disable automatic recording stop.

*-S*:

(uppercase S)

Requests a status information packet. The answer will be of the form:

```
connected: 1
connecting: 0
recording: 1
signal present: 1
signal absent: 0
stream seconds: 10
stream kBytes: 164
record seconds: 38
record kBytes: 3581
volume left: -6.2
volume right: -6.2
song: Stanley Clarke Trio - Under the Bridge
record path: /home/butt/recordings/recording_20220312.flac
listeners: 256
```

*-a <address>*:

Use this option to control a BUTT instance that is running on a remote computer.

In order to control a remote BUTT instance the BUTT instance must have been

started with the *-A* option. The parameter *address* can be either a IP address or a hostname.

*-p <port>*:

This should be set to the same port that has been given to the BUTT instance you want to control.

By default the command will be sent to port 1256.

*-U*:

(uppercase U)

Use this option to send commands via UDP instead of TCP.

# Chapter 22. Uninstall

*MacOS:*

Delete the **butt.app** from your *Application* folder and remove the configuration file from */Users/<username>/.buttrc*

*Windows:*

Run the Uninstaller from the BUTT folder in your windows start menu.

*Linux/MinGW:*

Run `sudo make uninstall` from the source tree and remove the configuration file from */home/<username>/.buttrc*

# Chapter 23. Contact

butt at danielnoethen dot de

# Chapter 24. Donate

Paypal: [paypal@danielnoethen.de](mailto:paypal@danielnoethen.de)

Patreon: [Become a Patron](#)

Apple Pay, Google Pay and more: [Donorbox](#)

*Cryptocurrencies:*

Bitcoin: bc1q4uq7h464rsu2cudrmuuqmc4tcr98d0edrhe5au

Litecoin: Ld9gntf8fsYpmVcbstFkzz5R3sNPC3AhTx

Monero:

85u8DacasxPNvKzY5kEiprBnbydDqg26yGAVEw7mdwccNFsrXmwCE4VQnV2JVfh5BTRheNnpDJqYj  
bqPrVRLEPAKP3dsYgc

If you want to use a different cryptocurrency please contact me.