

# Open Sound Control (OSC) for silentjack

## Contents

<b>1</b>	<b>Prerequisites</b>	<b>1</b>
1.1	JACK Audio Connection Kit . . . . .	1
1.2	liblo - Lightweight OSC Implementation . . . . .	1
<b>2</b>	<b>Installing silentjack (OSC)</b>	<b>1</b>
<b>3</b>	<b>Using silentjack (OSC)</b>	<b>1</b>
3.1	OSC Settings . . . . .	2
<b>4</b>	<b>Messages sent by silentjack</b>	<b>3</b>
4.1	/silentjack/started . . . . .	3
4.2	/silentjack/not_connected . . . . .	3
4.3	/silentjack/connected . . . . .	3
4.4	/silentjack/level . . . . .	4
4.5	/silentjack/silent . . . . .	4
4.6	/silentjack/run_cmd . . . . .	4
4.7	/silentjack/grace . . . . .	5
4.8	/silentjack/settings . . . . .	5
4.9	/silentjack/quit . . . . .	6
<b>5</b>	<b>Messages received by silentjack</b>	<b>6</b>
5.1	/silentjack/get_settings . . . . .	6
5.2	/silentjack/set_trigger_level . . . . .	6
5.3	/silentjack/set_silence_period . . . . .	7
5.4	/silentjack/set_grace_period . . . . .	7
5.5	/silentjack/set_verbose . . . . .	7
5.6	/silentjack/quit . . . . .	8
<b>6</b>	<b>Test it</b>	<b>8</b>
6.1	Application Notes . . . . .	8
<b>7</b>	<b>Download Links</b>	<b>8</b>

---

Last Update: 2013-06-28 22:06:30 +0200

The program **silentjack** can be used to listen to **JACK** ports and detect if the audio signal is below or above a defined level (dB) for a defined time interval. A fork of silentjack was made in order to sense and control the program via the UDP protocol **OSC**. The library **liblo** is used to implement OSC in silentjack.

Project home:

- [https://github.com/7890/silentjack\\_osc](https://github.com/7890/silentjack_osc)

silentjack OSC is a derivate of "silentjack - Silence / Dead Air Detector":

- <http://www.aelius.com/njh/silentjack/>

## 1 Prerequisites

### 1.1 JACK Audio Connection Kit

<http://www.jackaudio.org>

### 1.2 liblo - Lightweight OSC Implementation

```
git clone git://liblo.git.sourceforge.net/gitroot/liblo/liblo
```

See example programs `oscsend` and `oscdump`

A small modification will make the `oscdump` tool ready for shell hacks. <https://gist.github.com/7890/8d83f887b9dc771dcb00>

## 2 Installing silentjack (OSC)

If all prerequisites are met and the necessary build tools are available, it should be enough to do the following in a directory of choice:

```
$ git clone git://github.com/7890/silentjack_osc.git
$ cd silentjack_osc
$ ./autogen.sh && make && sudo make install
```

## 3 Using silentjack (OSC)

Many applications are possible without using the OSC features of silentjack simply by using the *run command on silence* feature.

However with the OSC modification, silentjack becomes even more powerful. Once started, it is a fully controllable network node that does inter-process communication (IPC) via OSC.

It provides a way of reconfiguring running instances of silentjack. As an example, the trigger level or silence period of a running silentjack instance could be configured by a scheduler. This allows to tailor sensing of a source to specific needs based on time (i.e. on a per-radio-show basis).

```
$ silentjack -h

silentjack version 0.3 OSC

Usage: silentjack [options] [COMMAND [ARG]...]
Options:  -c <port>    Connect to this port
          -n <name>    Name of this client (default 'silentjack')
          -l <db>      Trigger level (default -40 decibels)
          -p <secs>    Period of silence required (default 1 second)
          -d <db>      No-dynamic trigger level (default disabled)
          -P <secs>    No-dynamic period (default 10 seconds)
          -g <secs>    Grace period (default 0 seconds)
          -v           Enable verbose mode
          -q           Enable quiet mode
          -o <port>    Set OSC port for listening (default 7777, ? for random)
          -H <host>    Set OSC host to send to (default 127.0.0.1)
          -O <port>    Set OSC port to send to (default 7778)
          -X           Disable all OSC functions
          -h           Show help
```

### 3.1 OSC Settings

The following three settings define how OSC messages are processed.

- Port to start OSC Server and listen for incoming messages
- Host and port to send outgoing messages to (report to this address)

The default settings are

- Start OSC Server on port 7777
- Report to 127.0.0.1, port 7778

Example command line call with alternative OSC settings

```
$ silentjack -o 9999 -H 10.10.10.255 -O 10000
```

---

#### Note

The OSC *telling and answering* mechanism is different from other implementations

- Messages are not sent back to the host and port of the requester by default
- There is no pub/sub mechanism and therefore no subscribers list to serve

This allows the following scenarios

- A requesting process can be decoupled from the node that receives the answer
- Address of report host can be a broadcast address to serve all on subnet

---

There is a separate `-V` (verbose) command line switch for OSC.

Most of the settings like silence/grace period, trigger level, verbosity can be read and set while the program is running via OSC.

---

## 4 Messages sent by silentjack

### 4.1 /silentjack/started

Message Pattern:  
    /silentjack/started ssiifi

Parameters:  
    s: jack\_client\_name  
    s: osc\_server\_port  
    i: silence\_period           [sec]  
    i: grace\_period            [sec]  
    f: trigger\_level           [dB]  
    i: is\_verbose

Sent on event:  
    silentjack ready for service after startup.

Sent if:  
    (no restriction)

Example:  
    /silentjack/started ssiifi "silentjack" "7777" 5 0 -40.000000 1

### 4.2 /silentjack/not\_connected

Message Pattern:  
    /silentjack/not\_connected ss

Parameters:  
    s: jack\_client\_name  
    s: osc\_server\_port

Sent on event:  
    silentjack is not yet or not any longer connected to any jack source port.  
    Message is repeated every second.

Sent if:  
    verbose (`-V`)

Example:  
    /silentjack/not\_connected ss "silentjack" "7777"

### 4.3 /silentjack/connected

Message Pattern:  
    /silentjack/connected ss

Parameters:  
    s: jack\_client\_name  
    s: osc\_server\_port

Sent on event:  
    silentjack was just connected to a jack source port.  
    Message is sent once on connect and is followed by `/silentjack/level` messages.

Sent if:

```
verbose (`-V`)
```

Example:

```
/silentjack/connected ss "silentjack" "7777"
```

## 4.4 /silentjack/level

Message Pattern:

```
/silentjack/level ssiif
```

Parameters:

```
s: jack_client_name
s: osc_server_port
i: is_above_threshold
i: seconds_in_period    [sec]
f: level
```

Sent on event:

```
silentjack just made another evaluation of the audio signal.
This happens once per second and only if connected and
not in grace period.
```

Sent if:

```
verbose (`-V`)
```

Example:

```
/silentjack/level ssiif "silentjack" "7777" 0 4 -8.267532
```

## 4.5 /silentjack/silent

Message Pattern:

```
/silentjack/silent ssf
```

Parameters:

```
s: jack_client_name
s: osc_server_port
f: level                [dB]
```

Sent on event:

```
silentjack detected silence (max silence was reached).
```

Sent if:

```
verbose (`-V`)
```

Example:

```
/silentjack/silent ssf "silentjack" "7777" -45.267532
```

## 4.6 /silentjack/run\_cmd

Message Pattern:

```
/silentjack/run_cmd sss(s*)
```

Parameters:

```
s: jack_client_name
s: osc_server_port
s: command
```

```
(s: param 1)
(s: param n)
```

Sent on event:

silentjack runs a command (after silence detected) if silentjack was started with a command argument (last option).

Sent if:

(no restriction)

Example:

```
/silentjack/run_cmd sssss "silentjack" "7777" "/usr/local/bin/myscript.sh"
```

## 4.7 /silentjack/grace

Message Pattern:

```
/silentjack/grace ssi
```

Parameters:

```
s: jack_client_name
s: osc_server_port
i: seconds_in_grace_period      [sec]
```

Sent on event:

silentjack is in grace period.  
The message is repeated until the period is over.

Sent if:

verbose (`-V`)

Example:

```
/silentjack/grace ssi "silentjack" "7777" 9
```

## 4.8 /silentjack/settings

Message Pattern:

```
/silentjack/settings ssiifi
```

Parameters:

```
s: jack_client_name
s: osc_server_port
i: silence_period      [sec]
i: grace_period        [sec]
f: trigger_level       [dB]
i: is_verbose
```

Sent on event:

silentjack was reconfigured or requested to tell the current configuration.

Sent if:

(no restriction)

Example:

```
/silentjack/settings ssiifi "silentjack" "7777" 5 10 -40.000000 1
```

## 4.9 /silentjack/quit

Message Pattern:  
    /silentjack/quit ss

Parameters:  
    s: jack\_client\_name  
    s: osc\_server\_port

Sent on event:  
    silentjack is about to quit.  
    This is normally caused by CTRL+C or sending the message '/silentjack/quit'.

Sent if:  
    (no restriction)

Example:  
    /silentjack/quit ss "silentjack" "7777"

## 5 Messages received by silentjack



### Important

all get/set methods have a free-defined string as an additional argument (`req_id`). The string will be repeated in the requested answer from silentjack.

### 5.1 /silentjack/get\_settings

Message Pattern:  
    /silentjack/get\_settings s

Parameters:  
    s: req\_id

Action:  
    request silentjack to tell current settings.

Reply:  
    /silentjack/settings ssiifis

    s: jack\_client\_name  
    s: osc\_server\_port  
    i: silence\_period           [sec]  
    i: grace\_period            [sec]  
    f: trigger\_level           [dB]  
    i: is\_verbose  
    s: req\_id

### 5.2 /silentjack/set\_trigger\_level

Message Pattern:  
    /silentjack/set\_trigger\_level fs

Parameters:



```
f: trigger_level      [db]
s: req_id
```

Action:

request silentjack to reconfigure trigger level.

Reply:

/silentjack/settings ssiifis

### 5.3 /silentjack/set\_silence\_period

Message Pattern:

/silentjack/set\_silence\_period is

Parameters:

```
i: silence_period      [sec]
s: req_id
```

Action:

request silentjack to reconfigure silence period.

Reply:

/silentjack/settings ssiifis

### 5.4 /silentjack/set\_grace\_period

Message Pattern:

/silentjack/set\_grace\_period is

Parameters:

```
i: grace_period         [sec]
s: req_id
```

Action:

request silentjack to reconfigure grace period.

Reply:

/silentjack/settings ssiifis

### 5.5 /silentjack/set\_verbose

Message Pattern:

/silentjack/verbose is

Parameters:

```
i: is_verbose
s: req_id
```

Action:

request silentjack to be or not to be verbose OSC-wise (0/1).

Reply:

/silentjack/settings ssiifis

---

## 5.6 /silentjack/quit

```
Message Pattern:
    /silentjack/quit

No Parameters

Action:
    request silentjack to quit.

Reply:
    /silentjack/quit ss

    s: jack_client_name
    s: osc_server_port
```

## 6 Test it

```
$ silentjack -n "stream_silence_listener" -l -50 -p 30 -g 60 -V -c "mpg123-6153:left"
```

This will start an instance of silentjack in OSC verbose mode (-V) named `stream_silence_listener` (-n) and connect it to output port of mpg123 (-c) or any other JACK source of your choice. Since no specific arguments for OSC are given, the standard values are used: silentjack will receive messages on port 7777, and will report any events or answers to requests to localhost, port 7778.

If the audio level falls below -50 dB (-l) for at least 30 seconds (-p). When silence is detected, evaluation will start again after a grace period of 60 seconds (-g).

The `oscdump` program is an example client of `liblo` and can be found there. It will dump all messages sent by silentjack.

```
$ oscdump 7778
```

Messages to control a running silentjack can be sent for instance with `oscsend`.

```
$ oscsend localhost 7777 /stream_silence_listener/set_trigger_level fs -10 "foo"
```

### 6.1 Application Notes

In order to detect a faulty audio signal in means of a missing left or right channel, use two instances of silentjack, one per channel. silentjack can be used for any audio routing logic that depends on available / not available signals in an automatic way, for instance to drive fade-in / fade-out by precedence rules over an array of inputs.

## 7 Download Links

Download this document (.pdf):



- [https://github.com/7890/silentjack\\_osc/blob/master/asciidoc/silentjack\\_osc.pdf?raw=true](https://github.com/7890/silentjack_osc/blob/master/asciidoc/silentjack_osc.pdf?raw=true)

Download silentjack\_osc (.zip):



- [https://github.com/7890/silentjack\\_osc/archive/master.zip](https://github.com/7890/silentjack_osc/archive/master.zip)
-