

# MOTU AVB Datastore API

MOTU AVB devices are equipped with a powerful API for hardware control and monitoring. This document covers API version 0.0.0.

## The Datastore

The device's parameters are stored in a key-value store called the *datastore*. All parameters of interest are exposed as a key in the datastore. Each key in the datastore is a series of */*-separated *path components*. For example, `mix/aux/7/eq/highshelf/freq` represents the 8th aux channel's highshelf frequency in the EQ effect. All indices are 0-based.

## JSON over HTTP Interface

The full datastore is accessible via [JSON \(http://en.wikipedia.org/wiki/JSON\)](http://en.wikipedia.org/wiki/JSON) over [HTTP \(http://en.wikipedia.org/wiki/HTTP\)](http://en.wikipedia.org/wiki/HTTP). This section describes the basics of the API through a series of example [curl \(http://curl.haxx.se/\)](http://curl.haxx.se/) commands.

### Basics

The device hosts HTTP access to the datastore through the `/datastore` path. To get the entire contents of the datastore as JSON, simply GET `/datastore`:

```
> curl <yourdevice.local.>/datastore
(the full datastore)
```

To get the contents of a subtree of the datastore, simply append that subtree to the URL. For example, to get all the settings related to the gate effect on channel 17:

```
> curl <yourdevice.local.>/datastore/mix/chan/16/gate
{"release":500.000000,
 "enable":0.000000,
 "attack":100.000000,
 "threshold":0.1}
```

Again, note that all indices are 0-based. To get the value for a single key, append the full datastore path to the datastore URL. The resulting JSON object will have a single value, under a key named "value". For example, to get the name of the third output bank:

```
> curl <yourdevice.local.>/datastore/ext/obank/2/name
{"value":"ADAT A"}
```

### Making changes to the datastore

Changes to the datastore are made with the HTTP PATCH verb. POST is also supported and behaves identically for clients that

do not support PATCH. The API for setting values mirrors that for getting them: clients can PATCH the datastore root, a subtree, or a single value. The data must be form-encoded, with a form field named "json" containing a JSON-encoded object with the key/value pairs to change. If you are setting a single value, the JSON object should have one key named "value".

## Some examples:

### Setting a single value

This command sets the name of the first channel of the third output bank:

```
> curl --data 'json={"value":"My favorite channel"}' \
<yourdevice.local.>/datastore/ext/obank/2/ch/0/name
```

### Setting multiple values on the same subtree

This command sets the names of the first and second channels in the third output bank:

```
> curl --data 'json={"ch/0/name":"The Best of Channels", "ch/1/name":"The Worst of Channels"}' \
<yourdevice.local.>/datastore/ext/obank/2/
```

### Setting multiple values with full paths

This command sets the name of the first channel on the third output bank, and enables the gate effect on the first mixer channel.

```
> curl --data 'json={"ext/obank/2/ch/0/name":"I guess this channel is fine", "mix/chan/0/gate/enable":1}' \
<yourdevice.local.>/datastore
```

## ETags and Long Polling

The whole datastore has a [HTTP ETag \(http://en.wikipedia.org/wiki/HTTP\\_ETag\)](http://en.wikipedia.org/wiki/HTTP_ETag) representing the number of times the datastore has changed since boot. Each time a parameter is changed, this global ETag is incremented. For example, in this case the ETag is 5678:

```
> curl -s -D - <yourdevice.local.>/datastore -o /dev/null # only show headers
HTTP/1.1 200 OK
Connection: Keep-Alive
Transfer-Encoding: chunked
ETag: 5678
Content-Type: application/json
Cache-Control: no-cache
```

After the next change to the datastore, the ETag will be incremented to 5679.

To support long polling, the device has special behavior when the request includes an If-None-Match header. If the current

datastore ETag is newer (i.e., *greater in number*) than the sent If-None-Match ETag, the device will respond immediately.

```
> curl -H "If-None-Match: 5670" <yourdevice.local.>/datastore/ext/obank/2/name  
{"value":"ADAT A"}
```

If the If-None-Match ETag is equal to the current ETag, the device will not respond for 15 seconds. If 15 seconds elapse without a change, it will respond with 304 Not Modified.

However, if the datastore changes during the 15 second wait period, the device will immediately respond with all changes since the ETag passed in the If-None-Match header. This combination of behaviors enables clients to be notified of changes with low latency and a low polling frequency.

## The Client ID

Additionally, clients may pass in a client ID in a query string variable named "client". The client ID must be a number representable by a 32-bit unsigned integer (i.e., in the range \$0\$ to  $2^{32} - 1$ ). Datastore changes made by PATCH and POST requests with a given client ID will be filtered out of all long polling GET requests with the same client ID. This may be convenient for clients which do not wait for a round-trip before changing the user-visible UI. We recommend choosing a random integer in this range and using that as your client ID for the duration of your session.

### Example:

```
> curl <yourdevice.local.>/datastore?client=1479701624
```

## Datastore Types

Each datastore path has an assigned type. Each PUT or POST to a path must contain data that matches the type for that path.

- string a utf8 string, with '\', '"', and control codes escaped with a '\', according to the [JSON spec](http://en.wikipedia.org/wiki/JSON) (<http://en.wikipedia.org/wiki/JSON>).
- real a floating point number
- int an integer
- semver a [semver](http://semver.org/) (<http://semver.org/>) version string, e.g. 1.0.6+1234

Any type can be modified by the following "type modifiers" by appending <\_modifier> to the type:

- list a string containing colon separated list of objects - note that e.g. int\_list is represented by string, but each component must be convertible to an integer.
- pair a string containing a colon-separated pair of objects.
- opt an optional object (i.e. the object may not exist in the datastore)
- bool This is a special modifier that means 0 indicates "false", while any other value indicates "true".
- enum A special modifier that indicates the path can only take one of a finite number of values. The potential values are documented along with the path.

## Datastore Permissions

Each datastore path has a permission: either 'r' (read) or 'rw' (read/write). Clients can only change parameters marked 'rw'.

# Versioning

Each section of datastore parameters has a separate semver version associated with it. For each section, the current datastore version for that section lives in `ext/caps/<section>`. For example, the version for the `avb` section appears at `ext/caps/avb`. If the version path doesn't exist, that section does not exist on the device.

Each path is documented with the first version in which it appeared. Any other compatibility notes are mentioned in the description section. In keeping with the semver description, any breaking change will result in an increment of the major version, while non-breaking changes such as feature additions will cause the minor version to increment.

The HTTP protocol used to query datastore paths and the sections in the "global" section are both versioned by an "apiversion" parameter which lives outside the Datastore API. The easiest way to check this number is by a GET request to `/apiversion`.

```
> curl <yourdevice.local>/apiversion
0.0.0
```

This documentation applies specifically to global API versions equal to or above 0.0.0 and below 1.0.0.

## Datastore Path Placeholders

Many datastore paths are documented with certain components replaced by placeholders in angle brackets (`<>`). Some of these placeholders can have different values depending on the exact model of device, and are subject to change even in minor versions. For mixer and i/o parameters in particular, make sure you do a full datastore request first to see exactly which paths are available on your particular device.

## Global Settings

### uid

- Type: string
- Permission: r
- Available since global version: 0.0.0
- Description: The UID of the device. The UID is a 16 digit hexadecimal string that uniquely identifies this device on AVB networks.

### ext/caps/avb

- Type: semver\_opt
- Permission: r
- Available since global version: 0.0.0
- Description: The version of the avb section. If this path is absent, the device does not have the paths in the avb section.

### ext/caps/router

- Type: semver\_opt
- Permission: r
- Available since global version: 0.0.0
- Description: The version of the router section. If this path is absent, the device does not have the paths in the router section.

### ext/caps/mixer

- Type: semver\_opt
- Permission: r
- Available since global version: 0.0.0
- Description: The version of the mixer section. If this path is absent, the device does not have the paths in the mixer section.

## AVB (Audio Video Bridging) Settings

The avb section of the datastore is special because it includes information on all AVB devices in the target device's AVB network, in addition to the local parameters of that device. The list of all devices exists at avb/devs. Each device in that list maintains a separate subtree, containing all AVB parameters, located at avb/<uid>. Any AVB-capable device -- even those not created by MOTU -- will appear in the avb section, although MOTU-only parameters such as apiversion and url will only appear for MOTU devices.

### avb/devs

- Type: string\_list
- Permission: r
- Available since avb version: 0.0.0
- Description: A list of UIDs for AVB devices on the same network as this device.

### avb/<uid>/entity\_model\_id\_h32

- Type: int
- Permission: r
- Available since avb version: 0.0.0
- Description: The vendor id of the connected AVB device.

### avb/<uid>/entity\_model\_id\_l32

- Type: int
- Permission: r
- Available since avb version: 0.0.0
- Description: The model id of the connected AVB device.

### avb/<uid>/entity\_name

- Type: string
- Permission: rw
- Available since avb version: 0.0.0
- Description: The human readable name of the connected AVB device. On MOTU devices, this may be changed by the user or an API client (e.g., "My 1248").

### avb/<uid>/model\_name

- Type: string
- Permission: r
- Available since avb version: 0.0.0
- Description: The human readable model name of the connected AVB device (e.g., "1248").

### avb/<uid>/hostname

- Type: string\_opt
- Permission: r
- Available since avb version: 0.0.0
- Description: The sanitized hostname assigned to this device. This is only valid for MOTU devices. This may be different from entity\_name in that it won't have spaces or non-ascii characters (e.g., "My-1248").

## **avb/<uid>/master\_clock/capable**

- Type: int\_bool
- Permission: r
- Available since avb version: 0.0.0
- Description: True if this device supports MOTU Master Clock. MOTU Master Clock is a set of special datastore keys in the avb section that allows one device to quickly become the clock source of many others.

## **avb/<uid>/master\_clock/uid**

- Type: string\_opt
- Permission: rw
- Available since avb version: 0.0.0
- Description: The UID of the device the master\_clock stream is connected to, or the empty string if there is no connection. Only available for devices that are Master Clock capable (see master\_clock/capable above).

## **avb/<uid>/vendor\_name**

- Type: string
- Permission: r
- Available since avb version: 0.0.0
- Description: The human readable vendor name of the connected AVB device (e.g., "MOTU").

## **avb/<uid>/firmware\_version**

- Type: string
- Permission: r
- Available since avb version: 0.0.0
- Description: The human readable firmware version number of the connected AVB device. For MOTU devices, this will be a semver.

## **avb/<uid>/serial\_number**

- Type: string
- Permission: r
- Available since avb version: 0.0.0
- Description: The human readable serial number of the connected AVB device.

## **avb/<uid>/controller\_ignore**

- Type: int\_bool
- Permission: r
- Available since avb version: 0.0.0
- Description: True if this device should be ignored. If true, clients should not show this device in their UI.

## **avb/<uid>/acquired\_id**

- Type: string
- Permission: r
- Available since avb version: 0.0.0
- Description: The controller UID of the controller that acquired this box, or the empty string if no controller has acquired it. Acquisition is a part of the AVB standard that allows a controller to prevent other controllers from making changes on this device. You cannot initiate an acquisition from the datastore API, but you should avoid making changes on a device that has been acquired elsewhere.

## **avb/<uid>/motu.mdns.type**

- Type: string\_opt
- Permission: r
- Available since avb version: 0.0.0
- Description: The name of the device family for this device (e.g., "netiodevice"). This path is only valid for MOTU devices.

### **avb/<uid>/apiversion**

- Type: semver\_opt
- Permission: r
- Available since avb version: 0.0.0
- Description: The global datastore API version of the device. This path is only valid for MOTU devices.

### **avb/<uid>/url**

- Type: string\_opt
- Permission: r
- Available since avb version: 0.0.0
- Description: The canonical url of the device. This path is only valid for MOTU devices.

### **avb/<uid>/current\_configuration**

- Type: int
- Permission: rw
- Available since avb version: 0.0.0
- Description: The index of the currently active device configuration. MOTU devices only have one configuration, index 0. Other devices may have multiple available configurations.

### **avb/<uid>/cfg/<index>/object\_name**

- Type: string
- Permission: r
- Available since avb version: 0.0.0
- Description: The name of the configuration with the given index.

### **avb/<uid>/cfg/<index>/identify**

- Type: int\_bool
- Permission: rw
- Available since avb version: 0.0.0
- Description: True if the configuration is in identify mode. What identify mode means depends on the device. For MOTU devices, identify will flash the front panel backlight.

### **avb/<uid>/cfg/<index>/current\_sampling\_rate**

- Type: int
- Permission: rw
- Available since avb version: 0.0.0
- Description: The sampling rate of the configuration with the given index.

### **avb/<uid>/cfg/<index>/sample\_rates**

- Type: int\_list
- Permission: r
- Available since avb version: 0.0.0
- Description: A list of allowed sample rates for the configuration with the given index.

### **avb/<uid>/cfg/<index>/clock\_source\_index**

- Type: int
- Permission: rw
- Available since avb version: 0.0.0
- Description: The currently chosen clock source for the configuration with the given index.

### **avb/<uid>/cfg/<index>/clock\_sources/num**

- Type: int
- Permission: r
- Available since avb version: 0.0.0
- Description: The number of available clock sources for the given configuration.

### **avb/<uid>/cfg/<index>/clock\_sources/<index>/object\_name**

- Type: string
- Permission: r
- Available since avb version: 0.0.0
- Description: The name of the clock source with the given index.

### **avb/<uid>/cfg/<index>/clock\_sources/<index>/type**

- Type: string
- Permission: r
- Available since avb version: 0.0.0
- Description: The type of the clock source with the given index. The value will be one of "internal", "external", or "stream".

### **avb/<uid>/cfg/<index>/clock\_sources/<index>/stream\_id**

- Type: int\_opt
- Permission: r
- Available since avb version: 0.0.0
- Description: If the type of the clock source is "stream", the id of the stream from which it derives its clock. This path is only valid if the clock is a stream.

### **avb/<uid>/cfg/<index>/<input\_or\_output>\_streams/num**

- Type: int
- Permission: r
- Available since avb version: 0.0.0
- Description: The number of available input or output AVB streams.

### **avb/<uid>/cfg/<index>/<input\_or\_output>\_streams/<index>/object\_name**

- Type: string
- Permission: r
- Available since avb version: 0.0.0
- Description: The name of the input or output stream with the given index

### **avb/<uid>/cfg/<index>/<input\_or\_output>\_streams/<index>/num\_ch**

- Type: int
- Permission: r
- Available since avb version: 0.0.0
- Description: The number of channels on the input or output stream.

### **avb/<uid>/cfg/<index>/input\_streams/<index>/talker**

- Type: string\_pair



- Permission: rw
- Available since avb version: 0.0.0
- Description: The talker for the given input stream. The first element of the pair is the device UID, the second element of the pair is the stream ID that this stream is connected to.

### **ext/clockLocked**

- Type: int\_bool
- Permission: r
- Available since avb version: 0.0.0
- Description: True if the clock is locked.

## **Routing and I/O Settings**

### **ext/wordClockMode**

- Type: string
- Permission: rw
- Available since router version: 0.2.0
- Description: "1x" if the word clock out should always be a 1x rate or "follow" if it should always follow the system clock

### **ext/wordClockThru**

- Type: string
- Permission: rw
- Available since router version: 0.2.0
- Description: "thru" if the word clock output should be the same as the word clock input or "out" if it should be determined by the system clock

### **ext/smuxPerBank**

- Type: int\_bool
- Permission: r
- Available since router version: 0.2.0
- Description: True if each optical bank has its own SMUX setting

### **ext/vlimit/lookahead**

- Type: int\_bool\_opt
- Permission: rw
- Available since router version: 0.0.0
- Description: True if vLimit lookahead is enabled. vLimit lookahead provides better input limiting, at the cost of small amounts of extra latency. This path is only present on devices with access to vLimit.

### **ext/enableHostVolControls**

- Type: int\_bool
- Permission: rw
- Available since router version: 0.1.0
- Description: True if the computer is allowed to control the volumes of computer-to-device streams.

### **ext/<ibank\_or\_obank>/<index>/name**

- Type: string
- Permission: r
- Available since router version: 0.0.0
- Description: The name of the input or output bank

### **ext/<ibank\_or\_obank>/<index>/maxCh**

- Type: int
- Permission: r
- Available since router version: 0.0.0
- Description: The maximum possible number of channels in the input or output bank.

### **ext/<ibank\_or\_obank>/<index>/numCh**

- Type: int
- Permission: r
- Available since router version: 0.0.0
- Description: The number of channels available in this bank at its current sample rate.

### **ext/<ibank\_or\_obank>/<index>/userCh**

- Type: int
- Permission: rw
- Available since router version: 0.0.0
- Description: The number of channels that the user has enabled for this bank.

### **ext/<ibank\_or\_obank>/<index>/calcCh**

- Type: int
- Permission: r
- Available since router version: 0.0.0
- Description: The number of channels that are actually active. This is always the minimum of ext/<ibank\_or\_obank>/<index>/userCh and ext/<ibank\_or\_obank>/<index>/userCh.

### **ext/<ibank\_or\_obank>/<index>/smux**

- Type: string
- Permission: rw
- Available since router version: 0.2.0
- Description: For Optical banks, either "toslink" or "adat"

### **ext/ibank/<index>/madiClock**

- Type: string
- Permission: r
- Available since router version: 0.2.0
- Description: For MADi input banks, this is the 2x clock mode of the input stream-- "1x" for 48/44.1kHz frame clock, or "2x" for 88.2/96kHz frame clock

### **ext/obank/<index>/madiClock**

- Type: string
- Permission: rw
- Available since router version: 0.2.0
- Description: For MADi output banks, this is the 2x clock mode of the output stream-- "1x" for 48/44.1kHz frame clock, or "2x" for 88.2/96kHz frame clock

### **ext/ibank/<index>/madiFormat**

- Type: int
- Permission: r
- Available since router version: 0.2.0

- Description: 56 or 64 representing 56 or 64 MADi channels at 1x, 28 or 32 channels at 2x, or 14 or 16 channels at 4x, respectively

### **ext/obank/<index>/madiFormat**

- Type: int
- Permission: rw
- Available since router version: 0.2.0
- Description: 56 or 64 representing 56 or 64 MADi channels at 1x, 28 or 32 channels at 2x, or 14 or 16 channels at 4x, respectively

### **ext/<ibank\_or\_obank>/<index>/ch/<index>/name**

- Type: string
- Permission: rw
- Available since router version: 0.0.0
- Description: The channel's name.

### **ext/obank/<index>/ch/<index>/src**

- Type: int\_pair\_opt
- Permission: rw
- Available since router version: 0.0.0
- Description: If the output channel is connected to an input bank, a ":" separated pair in the form "":  
", otherwise, if unrouted, an empty string.

### **ext/<ibank\_or\_obank>/<index>/ch/<index>/phase**

- Type: int\_bool\_opt
- Permission: rw
- Available since router version: 0.0.0
- Description: True if the signal has its phase inverted. This is only applicable to some input or output channels.

### **ext/<ibank\_or\_obank>/<index>/ch/<index>/pad**

- Type: int\_bool\_opt
- Permission: rw
- Available since router version: 0.0.0
- Description: True if the 20 dB pad is engaged. This is only applicable to some input or output channels.

### **ext/ibank/<index>/ch/<index>/48V**

- Type: int\_bool\_opt
- Permission: rw
- Available since router version: 0.0.0
- Description: True if the 48V phantom power is engaged. This is only applicable to some input channels.

### **ext/ibank/<index>/ch/<index>/vLimit**

- Type: int\_bool\_opt
- Permission: rw
- Available since router version: 0.0.0
- Description: True if the vLimit limiter is engaged. This is only applicable to some input channels.

### **ext/ibank/<index>/ch/<index>/vClip**

- Type: int\_bool\_opt

- Permission: rw
- Available since router version: 0.0.0
- Description: True if vLimit clip is engaged. This is only applicable to some input channels.

### **ext/<ibank\_or\_obank>/<index>/ch/<index>/trim**

- Type: int\_opt
- Permission: rw
- Available since router version: 0.0.0
- Description: A dB-value for how much to trim this input or output channel. The range of this parameter is indicated by ext/<ibank\_or\_obank>/<index>/ch/<index>/trimRange. Only available for certain input or output channels.

### **ext/<ibank\_or\_obank>/<index>/ch/<index>/trimRange**

- Type: int\_pair\_opt
- Permission: rw
- Available since router version: 0.0.0
- Description: A pair of the minimum followed by maximum values allowed for the trim parameter on the input or output channel.

### **ext/<ibank\_or\_obank>/<index>/ch/<index>/stereoTrim**

- Type: int\_opt
- Permission: rw
- Available since router version: 0.0.0
- Description: A dB-value for how much to trim this input or output channel. This stereo trim affect both this channel and the next one. The range of this parameter is indicated by ext/<ibank\_or\_obank>/<index>/ch/<index>/stereoTrimRange. Only available for certain input or output channels.

### **ext/<ibank\_or\_obank>/<index>/ch/<index>/stereoTrimRange**

- Type: int\_pair\_opt
- Permission: rw
- Available since router version: 0.0.0
- Description: A pair of the minimum followed by maximum values allowed for the stereoTrim parameter on the input or output channel.

### **ext/<ibank\_or\_obank>/<index>/ch/<index>/connection**

- Type: int\_bool\_opt
- Permission: r
- Available since router version: 0.0.0
- Description: True if the channel has a physical connector plugged in (e.g., an audio jack). This information may not be available for all banks or devices.

## **Mixer Settings**

The mixer section as described is only valid for the current mixer version, 1.0. In future versions, paths, types, or valid parameter ranges may change.

### **mix/ctrls/dsp/usage**

- Type: int
- Permission: r
- Available since mixer version: 1.0.0
- Description: The approximate percentage of DSP resources used for mixing and effects.

### **mix/ctrls/<effect\_resource>/avail**

- Type: int\_bool\_opt
- Permission: r
- Available since mixer version: 1.0.0
- Description: True if there are enough DSP resources to enable one more of the given effect.

### **mix/chan/<index>/matrix/aux/<index>/send**

- Type: real
- Permission: rw
- Available since mixer version: 1.0.0
- Minimum Value: 0
- Maximum Value: 4
- Unit: linear

### **mix/chan/<index>/matrix/group/<index>/send**

- Type: real
- Permission: rw
- Available since mixer version: 1.0.0
- Minimum Value: 0
- Maximum Value: 4
- Unit: linear

### **mix/chan/<index>/matrix/reverb/<index>/send**

- Type: real
- Permission: rw
- Available since mixer version: 1.0.0
- Minimum Value: 0
- Maximum Value: 4
- Unit: linear

### **mix/chan/<index>/hpf/enable**

- Type: real\_bool
- Permission: rw
- Available since mixer version: 1.0.0

### **mix/chan/<index>/hpf/freq**

- Type: int
- Permission: rw
- Available since mixer version: 1.0.0
- Minimum Value: 20
- Maximum Value: 20000
- Unit: Hz

### **mix/chan/<index>/eq/highshelf/enable**

- Type: real\_bool
- Permission: rw
- Available since mixer version: 1.0.0

### **mix/chan/<index>/eq/highshelf/freq**

- Type: int
- Permission: rw
- Available since mixer version: 1.0.0
- Minimum Value: 20
- Maximum Value: 20000
- Unit: Hz

### **mix/chan/<index>/eq/highshelf/gain**

- Type: real
- Permission: rw
- Available since mixer version: 1.0.0
- Minimum Value: -20
- Maximum Value: 20
- Unit: dB

### **mix/chan/<index>/eq/highshelf/bw**

- Type: real
- Permission: rw
- Available since mixer version: 1.0.0
- Minimum Value: 0.01
- Maximum Value: 3
- Unit: octaves

### **mix/chan/<index>/eq/highshelf/mode**

- Type: real\_enum
- Permission: rw
- Available since mixer version: 1.0.0
- Possible Values: Shelf=0,Para=1

### **mix/chan/<index>/eq/mid1/enable**

- Type: real\_bool
- Permission: rw
- Available since mixer version: 1.0.0

### **mix/chan/<index>/eq/mid1/freq**

- Type: int
- Permission: rw
- Available since mixer version: 1.0.0
- Minimum Value: 20
- Maximum Value: 20000
- Unit: Hz

### **mix/chan/<index>/eq/mid1/gain**

- Type: real
- Permission: rw
- Available since mixer version: 1.0.0
- Minimum Value: -20
- Maximum Value: 20
- Unit: dB

### **mix/chan/<index>/eq/mid1/bw**

- Type: real
- Permission: rw
- Available since mixer version: 1.0.0
- Minimum Value: 0.01
- Maximum Value: 3
- Unit: octaves

### **mix/chan/<index>/eq/mid2/enable**

- Type: real\_bool
- Permission: rw
- Available since mixer version: 1.0.0

### **mix/chan/<index>/eq/mid2/freq**

- Type: int
- Permission: rw
- Available since mixer version: 1.0.0
- Minimum Value: 20
- Maximum Value: 20000
- Unit: Hz

### **mix/chan/<index>/eq/mid2/gain**

- Type: real
- Permission: rw
- Available since mixer version: 1.0.0
- Minimum Value: -20
- Maximum Value: 20
- Unit: dB

### **mix/chan/<index>/eq/mid2/bw**

- Type: real
- Permission: rw
- Available since mixer version: 1.0.0
- Minimum Value: 0.01
- Maximum Value: 3
- Unit: octaves

### **mix/chan/<index>/eq/lowshelf/enable**

- Type: real\_bool
- Permission: rw
- Available since mixer version: 1.0.0

### **mix/chan/<index>/eq/lowshelf/freq**

- Type: int
- Permission: rw
- Available since mixer version: 1.0.0
- Minimum Value: 20
- Maximum Value: 20000
- Unit: Hz

### **mix/chan/<index>/eq/lowshelf/gain**

- Type: real
- Permission: rw
- Available since mixer version: 1.0.0
- Minimum Value: -20
- Maximum Value: 20
- Unit: dB

### **mix/chan/<index>/eq/lowshelf/bw**

- Type: real
- Permission: rw
- Available since mixer version: 1.0.0
- Minimum Value: 0.01
- Maximum Value: 3
- Unit: octaves

### **mix/chan/<index>/eq/lowshelf/mode**

- Type: real\_enum
- Permission: rw
- Available since mixer version: 1.0.0
- Possible Values: Shelf=0,Para=1

### **mix/chan/<index>/gate/enable**

- Type: real\_bool
- Permission: rw
- Available since mixer version: 1.0.0

### **mix/chan/<index>/gate/release**

- Type: real
- Permission: rw
- Available since mixer version: 1.0.0
- Minimum Value: 50
- Maximum Value: 2000
- Unit: ms

### **mix/chan/<index>/gate/threshold**

- Type: real
- Permission: rw
- Available since mixer version: 1.0.0
- Minimum Value: 0
- Maximum Value: 1
- Unit: linear

### **mix/chan/<index>/gate/attack**

- Type: real
- Permission: rw
- Available since mixer version: 1.0.0
- Minimum Value: 10
- Maximum Value: 500
- Unit: ms

### **mix/chan/<index>/comp/enable**



- Type: real\_bool
- Permission: rw
- Available since mixer version: 1.0.0

### **mix/chan/<index>/comp/release**

- Type: real
- Permission: rw
- Available since mixer version: 1.0.0
- Minimum Value: 10
- Maximum Value: 2000
- Unit: ms

### **mix/chan/<index>/comp/threshold**

- Type: real
- Permission: rw
- Available since mixer version: 1.0.0
- Minimum Value: -40
- Maximum Value: 0
- Unit: dB

### **mix/chan/<index>/comp/ratio**

- Type: real
- Permission: rw
- Available since mixer version: 1.0.0
- Minimum Value: 1
- Maximum Value: 10

### **mix/chan/<index>/comp/attack**

- Type: real
- Permission: rw
- Available since mixer version: 1.0.0
- Minimum Value: 10
- Maximum Value: 100
- Unit: ms

### **mix/chan/<index>/comp/trim**

- Type: real
- Permission: rw
- Available since mixer version: 1.0.0
- Minimum Value: -20
- Maximum Value: 20
- Unit: dB

### **mix/chan/<index>/comp/peak**

- Type: real\_enum
- Permission: rw
- Available since mixer version: 1.0.0
- Possible Values: RMS=0,Peak=1

### **mix/chan/<index>/matrix/enable**

- Type: real\_bool
- Permission: rw
- Available since mixer version: 1.0.0

### **mix/chan/<index>/matrix/solo**

- Type: real\_bool
- Permission: rw
- Available since mixer version: 1.0.0

### **mix/chan/<index>/matrix/mute**

- Type: real\_bool
- Permission: rw
- Available since mixer version: 1.0.0

### **mix/chan/<index>/matrix/pan**

- Type: real
- Permission: rw
- Available since mixer version: 1.0.0
- Minimum Value: -1
- Maximum Value: 1
- Unit: pan

### **mix/chan/<index>/matrix/fader**

- Type: real
- Permission: rw
- Available since mixer version: 1.0.0
- Minimum Value: 0
- Maximum Value: 4
- Unit: linear

### **mix/main/<index>/eq/highshelf/enable**

- Type: real\_bool
- Permission: rw
- Available since mixer version: 1.0.0

### **mix/main/<index>/eq/highshelf/freq**

- Type: int
- Permission: rw
- Available since mixer version: 1.0.0
- Minimum Value: 20
- Maximum Value: 20000
- Unit: Hz

### **mix/main/<index>/eq/highshelf/gain**

- Type: real
- Permission: rw
- Available since mixer version: 1.0.0
- Minimum Value: -20
- Maximum Value: 20
- Unit: dB

### **mix/main/<index>/eq/highshelf/bw**

- Type: real
- Permission: rw
- Available since mixer version: 1.0.0
- Minimum Value: 0.01
- Maximum Value: 3
- Unit: octaves

### **mix/main/<index>/eq/highshelf/mode**

- Type: real\_enum
- Permission: rw
- Available since mixer version: 1.0.0
- Possible Values: Shelf=0,Para=1

### **mix/main/<index>/eq/mid1/enable**

- Type: real\_bool
- Permission: rw
- Available since mixer version: 1.0.0

### **mix/main/<index>/eq/mid1/freq**

- Type: int
- Permission: rw
- Available since mixer version: 1.0.0
- Minimum Value: 20
- Maximum Value: 20000
- Unit: Hz

### **mix/main/<index>/eq/mid1/gain**

- Type: real
- Permission: rw
- Available since mixer version: 1.0.0
- Minimum Value: -20
- Maximum Value: 20
- Unit: dB

### **mix/main/<index>/eq/mid1/bw**

- Type: real
- Permission: rw
- Available since mixer version: 1.0.0
- Minimum Value: 0.01
- Maximum Value: 3
- Unit: octaves

### **mix/main/<index>/eq/mid2/enable**

- Type: real\_bool
- Permission: rw
- Available since mixer version: 1.0.0

### **mix/main/<index>/eq/mid2/freq**

- Type: int
- Permission: rw
- Available since mixer version: 1.0.0
- Minimum Value: 20
- Maximum Value: 20000
- Unit: Hz

### **mix/main/<index>/eq/mid2/gain**

- Type: real
- Permission: rw
- Available since mixer version: 1.0.0
- Minimum Value: -20
- Maximum Value: 20
- Unit: dB

### **mix/main/<index>/eq/mid2/bw**

- Type: real
- Permission: rw
- Available since mixer version: 1.0.0
- Minimum Value: 0.01
- Maximum Value: 3
- Unit: octaves

### **mix/main/<index>/eq/lowshelf/enable**

- Type: real\_bool
- Permission: rw
- Available since mixer version: 1.0.0

### **mix/main/<index>/eq/lowshelf/freq**

- Type: int
- Permission: rw
- Available since mixer version: 1.0.0
- Minimum Value: 20
- Maximum Value: 20000
- Unit: Hz

### **mix/main/<index>/eq/lowshelf/gain**

- Type: real
- Permission: rw
- Available since mixer version: 1.0.0
- Minimum Value: -20
- Maximum Value: 20
- Unit: dB

### **mix/main/<index>/eq/lowshelf/bw**

- Type: real
- Permission: rw
- Available since mixer version: 1.0.0
- Minimum Value: 0.01
- Maximum Value: 3
- Unit: octaves

## **mix/main/<index>/eq/lowshelf/mode**

- Type: real\_enum
- Permission: rw
- Available since mixer version: 1.0.0
- Possible Values: Shelf=0,Para=1

## **mix/main/<index>/leveler/enable**

- Type: real\_bool
- Permission: rw
- Available since mixer version: 1.0.0

## **mix/main/<index>/leveler/makeup**

- Type: real
- Permission: rw
- Available since mixer version: 1.0.0
- Minimum Value: 0
- Maximum Value: 100
- Unit: %

## **mix/main/<index>/leveler/reduction**

- Type: real
- Permission: rw
- Available since mixer version: 1.0.0
- Minimum Value: 0
- Maximum Value: 100
- Unit: %

## **mix/main/<index>/leveler/limit**

- Type: real\_bool
- Permission: rw
- Available since mixer version: 1.0.0

## **mix/main/<index>/matrix/enable**

- Type: real\_bool
- Permission: rw
- Available since mixer version: 1.0.0

## **mix/main/<index>/matrix/mute**

- Type: real\_bool
- Permission: rw
- Available since mixer version: 1.0.0

## **mix/main/<index>/matrix/fader**

- Type: real
- Permission: rw
- Available since mixer version: 1.0.0
- Minimum Value: 0
- Maximum Value: 4
- Unit: linear

### **mix/aux/<index>/eq/highshelf/enable**

- Type: real\_bool
- Permission: rw
- Available since mixer version: 1.0.0

### **mix/aux/<index>/eq/highshelf/freq**

- Type: int
- Permission: rw
- Available since mixer version: 1.0.0
- Minimum Value: 20
- Maximum Value: 20000
- Unit: Hz

### **mix/aux/<index>/eq/highshelf/gain**

- Type: real
- Permission: rw
- Available since mixer version: 1.0.0
- Minimum Value: -20
- Maximum Value: 20
- Unit: dB

### **mix/aux/<index>/eq/highshelf/bw**

- Type: real
- Permission: rw
- Available since mixer version: 1.0.0
- Minimum Value: 0.01
- Maximum Value: 3
- Unit: octaves

### **mix/aux/<index>/eq/highshelf/mode**

- Type: real\_enum
- Permission: rw
- Available since mixer version: 1.0.0
- Possible Values: Shelf=0,Para=1

### **mix/aux/<index>/eq/mid1/enable**

- Type: real\_bool
- Permission: rw
- Available since mixer version: 1.0.0

### **mix/aux/<index>/eq/mid1/freq**

- Type: int
- Permission: rw
- Available since mixer version: 1.0.0
- Minimum Value: 20
- Maximum Value: 20000
- Unit: Hz

### **mix/aux/<index>/eq/mid1/gain**

- Type: real
- Permission: rw
- Available since mixer version: 1.0.0
- Minimum Value: -20
- Maximum Value: 20
- Unit: dB

### **mix/aux/<index>/eq/mid1/bw**

- Type: real
- Permission: rw
- Available since mixer version: 1.0.0
- Minimum Value: 0.01
- Maximum Value: 3
- Unit: octaves

### **mix/aux/<index>/eq/mid2/enable**

- Type: real\_bool
- Permission: rw
- Available since mixer version: 1.0.0

### **mix/aux/<index>/eq/mid2/freq**

- Type: int
- Permission: rw
- Available since mixer version: 1.0.0
- Minimum Value: 20
- Maximum Value: 20000
- Unit: Hz

### **mix/aux/<index>/eq/mid2/gain**

- Type: real
- Permission: rw
- Available since mixer version: 1.0.0
- Minimum Value: -20
- Maximum Value: 20
- Unit: dB

### **mix/aux/<index>/eq/mid2/bw**

- Type: real
- Permission: rw
- Available since mixer version: 1.0.0
- Minimum Value: 0.01
- Maximum Value: 3
- Unit: octaves

### **mix/aux/<index>/eq/lowshelf/enable**

- Type: real\_bool
- Permission: rw
- Available since mixer version: 1.0.0

### **mix/aux/<index>/eq/lowshelf/freq**

- Type: int
- Permission: rw
- Available since mixer version: 1.0.0
- Minimum Value: 20
- Maximum Value: 20000
- Unit: Hz

### **mix/aux/<index>/eq/lowshelf/gain**

- Type: real
- Permission: rw
- Available since mixer version: 1.0.0
- Minimum Value: -20
- Maximum Value: 20
- Unit: dB

### **mix/aux/<index>/eq/lowshelf/bw**

- Type: real
- Permission: rw
- Available since mixer version: 1.0.0
- Minimum Value: 0.01
- Maximum Value: 3
- Unit: octaves

### **mix/aux/<index>/eq/lowshelf/mode**

- Type: real\_enum
- Permission: rw
- Available since mixer version: 1.0.0
- Possible Values: Shelf=0,Para=1

### **mix/aux/<index>/matrix/enable**

- Type: real\_bool
- Permission: rw
- Available since mixer version: 1.0.0

### **mix/aux/<index>/matrix/prefader**

- Type: real\_bool
- Permission: rw
- Available since mixer version: 1.0.0

### **mix/aux/<index>/matrix/mute**

- Type: real\_bool
- Permission: rw
- Available since mixer version: 1.0.0

### **mix/aux/<index>/matrix/fader**

- Type: real
- Permission: rw
- Available since mixer version: 1.0.0
- Minimum Value: 0
- Maximum Value: 4



- Unit: linear

### **mix/group/<index>/matrix/aux/<index>/send**

- Type: real
- Permission: rw
- Available since mixer version: 1.0.0
- Minimum Value: 0
- Maximum Value: 4
- Unit: linear

### **mix/group/<index>/matrix/reverb/<index>/send**

- Type: real
- Permission: rw
- Available since mixer version: 1.0.0
- Minimum Value: 0
- Maximum Value: 4
- Unit: linear

### **mix/group/<index>/eq/highshelf/enable**

- Type: real\_bool
- Permission: rw
- Available since mixer version: 1.0.0

### **mix/group/<index>/eq/highshelf/freq**

- Type: int
- Permission: rw
- Available since mixer version: 1.0.0
- Minimum Value: 20
- Maximum Value: 20000
- Unit: Hz

### **mix/group/<index>/eq/highshelf/gain**

- Type: real
- Permission: rw
- Available since mixer version: 1.0.0
- Minimum Value: -20
- Maximum Value: 20
- Unit: dB

### **mix/group/<index>/eq/highshelf/bw**

- Type: real
- Permission: rw
- Available since mixer version: 1.0.0
- Minimum Value: 0.01
- Maximum Value: 3
- Unit: octaves

### **mix/group/<index>/eq/highshelf/mode**

- Type: real\_enum
- Permission: rw

- Available since mixer version: 1.0.0
- Possible Values: Shelf=0,Para=1

### **mix/group/<index>/eq/mid1/enable**

- Type: real\_bool
- Permission: rw
- Available since mixer version: 1.0.0

### **mix/group/<index>/eq/mid1/freq**

- Type: int
- Permission: rw
- Available since mixer version: 1.0.0
- Minimum Value: 20
- Maximum Value: 20000
- Unit: Hz

### **mix/group/<index>/eq/mid1/gain**

- Type: real
- Permission: rw
- Available since mixer version: 1.0.0
- Minimum Value: -20
- Maximum Value: 20
- Unit: dB

### **mix/group/<index>/eq/mid1/bw**

- Type: real
- Permission: rw
- Available since mixer version: 1.0.0
- Minimum Value: 0.01
- Maximum Value: 3
- Unit: octaves

### **mix/group/<index>/eq/mid2/enable**

- Type: real\_bool
- Permission: rw
- Available since mixer version: 1.0.0

### **mix/group/<index>/eq/mid2/freq**

- Type: int
- Permission: rw
- Available since mixer version: 1.0.0
- Minimum Value: 20
- Maximum Value: 20000
- Unit: Hz

### **mix/group/<index>/eq/mid2/gain**

- Type: real
- Permission: rw
- Available since mixer version: 1.0.0
- Minimum Value: -20

- Maximum Value: 20
- Unit: dB

### **mix/group/<index>/eq/mid2/bw**

- Type: real
- Permission: rw
- Available since mixer version: 1.0.0
- Minimum Value: 0.01
- Maximum Value: 3
- Unit: octaves

### **mix/group/<index>/eq/lowshelf/enable**

- Type: real\_bool
- Permission: rw
- Available since mixer version: 1.0.0

### **mix/group/<index>/eq/lowshelf/freq**

- Type: int
- Permission: rw
- Available since mixer version: 1.0.0
- Minimum Value: 20
- Maximum Value: 20000
- Unit: Hz

### **mix/group/<index>/eq/lowshelf/gain**

- Type: real
- Permission: rw
- Available since mixer version: 1.0.0
- Minimum Value: -20
- Maximum Value: 20
- Unit: dB

### **mix/group/<index>/eq/lowshelf/bw**

- Type: real
- Permission: rw
- Available since mixer version: 1.0.0
- Minimum Value: 0.01
- Maximum Value: 3
- Unit: octaves

### **mix/group/<index>/eq/lowshelf/mode**

- Type: real\_enum
- Permission: rw
- Available since mixer version: 1.0.0
- Possible Values: Shelf=0,Para=1

### **mix/group/<index>/leveler/enable**

- Type: real\_bool
- Permission: rw
- Available since mixer version: 1.0.0

### **mix/group/<index>/leveler/makeup**

- Type: real
- Permission: rw
- Available since mixer version: 1.0.0
- Minimum Value: 0
- Maximum Value: 100
- Unit: %

### **mix/group/<index>/leveler/reduction**

- Type: real
- Permission: rw
- Available since mixer version: 1.0.0
- Minimum Value: 0
- Maximum Value: 100
- Unit: %

### **mix/group/<index>/leveler/limit**

- Type: real\_bool
- Permission: rw
- Available since mixer version: 1.0.0

### **mix/group/<index>/matrix/enable**

- Type: real\_bool
- Permission: rw
- Available since mixer version: 1.0.0

### **mix/group/<index>/matrix/solo**

- Type: real\_bool
- Permission: rw
- Available since mixer version: 1.0.0

### **mix/group/<index>/matrix/prefader**

- Type: real\_bool
- Permission: rw
- Available since mixer version: 1.0.0

### **mix/group/<index>/matrix/mute**

- Type: real\_bool
- Permission: rw
- Available since mixer version: 1.0.0

### **mix/group/<index>/matrix/fader**

- Type: real
- Permission: rw
- Available since mixer version: 1.0.0
- Minimum Value: 0
- Maximum Value: 4
- Unit: linear

### **mix/reverb/<index>/matrix/aux/<index>/send**

- Type: real
- Permission: rw
- Available since mixer version: 1.0.0
- Minimum Value: 0
- Maximum Value: 4
- Unit: linear

### **mix/reverb/<index>/matrix/reverb/<index>/send**

- Type: real
- Permission: rw
- Available since mixer version: 1.0.0
- Minimum Value: 0
- Maximum Value: 4
- Unit: linear

### **mix/reverb/<index>/eq/highshelf/enable**

- Type: real\_bool
- Permission: rw
- Available since mixer version: 1.0.0

### **mix/reverb/<index>/eq/highshelf/freq**

- Type: int
- Permission: rw
- Available since mixer version: 1.0.0
- Minimum Value: 20
- Maximum Value: 20000
- Unit: Hz

### **mix/reverb/<index>/eq/highshelf/gain**

- Type: real
- Permission: rw
- Available since mixer version: 1.0.0
- Minimum Value: -20
- Maximum Value: 20
- Unit: dB

### **mix/reverb/<index>/eq/highshelf/bw**

- Type: real
- Permission: rw
- Available since mixer version: 1.0.0
- Minimum Value: 0.01
- Maximum Value: 3
- Unit: octaves

### **mix/reverb/<index>/eq/highshelf/mode**

- Type: real\_enum
- Permission: rw
- Available since mixer version: 1.0.0
- Possible Values: Shelf=0,Para=1

### **mix/reverb/<index>/eq/mid1/enable**

- Type: real\_bool
- Permission: rw
- Available since mixer version: 1.0.0

### **mix/reverb/<index>/eq/mid1/freq**

- Type: int
- Permission: rw
- Available since mixer version: 1.0.0
- Minimum Value: 20
- Maximum Value: 20000
- Unit: Hz

### **mix/reverb/<index>/eq/mid1/gain**

- Type: real
- Permission: rw
- Available since mixer version: 1.0.0
- Minimum Value: -20
- Maximum Value: 20
- Unit: dB

### **mix/reverb/<index>/eq/mid1/bw**

- Type: real
- Permission: rw
- Available since mixer version: 1.0.0
- Minimum Value: 0.01
- Maximum Value: 3
- Unit: octaves

### **mix/reverb/<index>/eq/mid2/enable**

- Type: real\_bool
- Permission: rw
- Available since mixer version: 1.0.0

### **mix/reverb/<index>/eq/mid2/freq**

- Type: int
- Permission: rw
- Available since mixer version: 1.0.0
- Minimum Value: 20
- Maximum Value: 20000
- Unit: Hz

### **mix/reverb/<index>/eq/mid2/gain**

- Type: real
- Permission: rw
- Available since mixer version: 1.0.0
- Minimum Value: -20
- Maximum Value: 20
- Unit: dB

### **mix/reverb/<index>/eq/mid2/bw**

- Type: real
- Permission: rw
- Available since mixer version: 1.0.0
- Minimum Value: 0.01
- Maximum Value: 3
- Unit: octaves

### **mix/reverb/<index>/eq/lowshelf/enable**

- Type: real\_bool
- Permission: rw
- Available since mixer version: 1.0.0

### **mix/reverb/<index>/eq/lowshelf/freq**

- Type: int
- Permission: rw
- Available since mixer version: 1.0.0
- Minimum Value: 20
- Maximum Value: 20000
- Unit: Hz

### **mix/reverb/<index>/eq/lowshelf/gain**

- Type: real
- Permission: rw
- Available since mixer version: 1.0.0
- Minimum Value: -20
- Maximum Value: 20
- Unit: dB

### **mix/reverb/<index>/eq/lowshelf/bw**

- Type: real
- Permission: rw
- Available since mixer version: 1.0.0
- Minimum Value: 0.01
- Maximum Value: 3
- Unit: octaves

### **mix/reverb/<index>/eq/lowshelf/mode**

- Type: real\_enum
- Permission: rw
- Available since mixer version: 1.0.0
- Possible Values: Shelf=0,Para=1

### **mix/reverb/<index>/leveler/enable**

- Type: real\_bool
- Permission: rw
- Available since mixer version: 1.0.0

### **mix/reverb/<index>/leveler/makeup**

- Type: real
- Permission: rw
- Available since mixer version: 1.0.0
- Minimum Value: 0
- Maximum Value: 100
- Unit: %

### **mix/reverb/<index>/leveler/reduction**

- Type: real
- Permission: rw
- Available since mixer version: 1.0.0
- Minimum Value: 0
- Maximum Value: 100
- Unit: %

### **mix/reverb/<index>/leveler/limit**

- Type: real\_bool
- Permission: rw
- Available since mixer version: 1.0.0

### **mix/reverb/<index>/matrix/enable**

- Type: real\_bool
- Permission: rw
- Available since mixer version: 1.0.0

### **mix/reverb/<index>/matrix/solo**

- Type: real\_bool
- Permission: rw
- Available since mixer version: 1.0.0

### **mix/reverb/<index>/matrix/prefader**

- Type: real\_bool
- Permission: rw
- Available since mixer version: 1.0.0

### **mix/reverb/<index>/matrix/mute**

- Type: real\_bool
- Permission: rw
- Available since mixer version: 1.0.0

### **mix/reverb/<index>/matrix/fader**

- Type: real
- Permission: rw
- Available since mixer version: 1.0.0
- Minimum Value: 0
- Maximum Value: 4
- Unit: linear

### **mix/reverb/<index>/reverb/enable**

- Type: real\_bool



- Permission: rw
- Available since mixer version: 1.0.0

### **mix/reverb/<index>/reverb/reverbtime**

- Type: int
- Permission: rw
- Available since mixer version: 1.0.0
- Minimum Value: 100
- Maximum Value: 60000
- Unit: ms

### **mix/reverb/<index>/reverb/hf**

- Type: int
- Permission: rw
- Available since mixer version: 1.0.0
- Minimum Value: 500
- Maximum Value: 15000
- Unit: Hz

### **mix/reverb/<index>/reverb/mf**

- Type: int
- Permission: rw
- Available since mixer version: 1.0.0
- Minimum Value: 500
- Maximum Value: 15000
- Unit: Hz

### **mix/reverb/<index>/reverb/predelay**

- Type: int
- Permission: rw
- Available since mixer version: 1.0.0
- Minimum Value: 0
- Maximum Value: 500
- Unit: ms

### **mix/reverb/<index>/reverb/mfratio**

- Type: int
- Permission: rw
- Available since mixer version: 1.0.0
- Minimum Value: 1
- Maximum Value: 100
- Unit: %

### **mix/reverb/<index>/reverb/hfratio**

- Type: int
- Permission: rw
- Available since mixer version: 1.0.0
- Minimum Value: 1
- Maximum Value: 100
- Unit: %

### **mix/reverb/<index>/reverb/tailsread**

- Type: int
- Permission: rw
- Available since mixer version: 1.0.0
- Minimum Value: -100
- Maximum Value: 100
- Unit: %

### **mix/reverb/<index>/reverb/mod**

- Type: int
- Permission: rw
- Available since mixer version: 1.0.0
- Minimum Value: 0
- Maximum Value: 100
- Unit: %

### **mix/monitor/<index>/matrix/enable**

- Type: real\_bool
- Permission: rw
- Available since mixer version: 1.0.0

### **mix/monitor/<index>/matrix/mute**

- Type: real\_bool
- Permission: rw
- Available since mixer version: 1.0.0

### **mix/monitor/<index>/matrix/fader**

- Type: real
- Permission: rw
- Available since mixer version: 1.0.0
- Minimum Value: 0
- Maximum Value: 4
- Unit: linear

### **mix/monitor/<index>/assign**

- Type: int
- Permission: rw
- Available since mixer version: 1.0.0
- Minimum Value: -2
- Maximum Value: 4096

### **mix/monitor/<index>/override**

- Type: int
- Permission: rw
- Available since mixer version: 1.0.0
- Minimum Value: -1
- Maximum Value: 4096

### **mix/monitor/<index>/auto**

- Type: real\_bool
- Permission: rw
- Available since mixer version: 1.0.0