

Remote control

When osc messages are received, the default behavior is to update the value of all widgets that match the address and preArgs. Additionally, Open Stage Control responds to some general osc commands.



Multi-client setup

These commands are interpreted by each client connected to the server so there might be more than one reply. Using a custom module is the only way to send a message to a single client.

`/EDIT id properties options`

Apply a set of options to an existing widget by replacing the old ones with the new ones. This ignores the widget's `lock` property.

- `id`: `string`, widget's `id`
- `properties`: `string`, [JSON5](#) stringified object defining the new properties to merge
- `example`: `{"label":"New Label", "color":"red"}`
- `options` (optional): `string`, [JSON5](#) stringified object defining extra option flags:
 - `noWarning`: set to `true` to prevent further warning when exiting.



Warning

Editing widgets is cpu expensive; for small and recurrent changes, consider using [osc listeners](#) instead.

`/EDIT/MERGE id properties options`

Apply a set of options to an existing widget by merging them to the widget's options.

/EDIT/UNDO

Undo editing action

/EDIT/REDO

Redo editing action

/EDIT/GET target id

Sends back a widget's data (JSON stringified object), including its children, to specified target.

- `target`: string, ip:port pair
- `id`: string, widget's id

Replies `/EDIT/GET id data`

- `id`: string
 - `data`: string
-

/EDIT/GET target address preArg1 preArg2 ...

Sends back a widget's data (JSON stringified object), including its children, to specified target.

- `target`: string, ip:port pair
- `address`: string, widget's address
- `preArg[1...]`: *, widget's preArgs

Replies `/EDIT/GET address preArg1 preArg2 ... data`

- `address`: string, widget's address
 - `preArg[1...]`: *, widget's preArgs
 - `data`: string
-

/GET target id

Sends back a widget's value to specified target.

- `target`: string, ip:port pair
- `id`: string, widget's id

Replies `/GET id value`

- `id`: string
 - `value`: *
-

`/GET target address preArg1 preArg2 ...`

Sends back a widget's value to specified target.

- `target`: string, ip:port pair
- `address`: string, widget's address
- `preArg[1...]`: *, widget's preArgs

Replies `/GET address preArg1 preArg2 ... value`

- `address`: string, widget's address
 - `preArg[1...]`: *, widget's preArgs
 - `value`: *
-

`/GET/#`

Same as `/GET` but uses the widget's address instead of `/GET` to reply.

`/NOTIFY icon message`

Sends a notification message to the client.

- `icon`: icon for the notification, for example: `^play`, `^pause`, etc (see <https://fontawesome.com/icons?d=gallery&s=solid&m=free>)
 - `message`: message content
-

`/LOG message`

Print message in the client's console.

`/SET id value`

Set a widget's value as if it was interacted with from the interface. This is likely to make it send its value.

- `id`: string, widget's `id`
- `value`: *, widget's new value

`/SET address preArg1 preArg2 ... value`

Set a widget's value as if it was interacted with from the interface. This is likely to make it send its value.

- `address`: string, widget's `address`
- `preArg[1...]`: *, widget's `preArgs`
- `value`: *, widget's new value

`/STATE/GET target`

Sends back the app's state to specified target

- `target`: string, `ip:port` pair

`/STATE/SET state`

Set widgets' state

- `state`: string, json stringified object (`"widget_id": value` pairs)

`/STATE/STORE`

Save the state of all widgets in the temporary slot.

`/STATE/RECALL`

Reload saved state from the temporary slot.

`/STATE/SEND`

Make all widgets send their current value.

`/STATE/OPEN path.state`

Open state file `path.state` .

`/STATE/SAVE path.state`

Save state session to `path.state` . If `path.state` is omitted, current session's path is used.

`/SESSION/OPEN path.json`

Open session file `path.json` .

`/SESSION/SAVE path.json`

Save current session to `path.json` . If `path.json` is omitted, current session's path is used.

`/TABS tab_id1 tab_id2 ...`

Open the tabs designated by the id parameters.

`/SCRIPT code`

Run code as if it were executed by a script widget.

`/RELOAD`

Reload page in the browser.