

OSC Source Management API

Context

This document describes the OSC write endpoints for creating and deleting sources from an external controller.

API Reading Instructions

- Replies are sent to the globally configured OSC output IP and output port.
- Replies are OSC bundles.
- Each returned property is one OSC message inside the bundle.
- Reply arguments are typed as `string`, `int32`, or `float32`.
- Booleans are sent and returned as `int32 0` or `1`.
- Source indexes are one-based top-level source-bin positions.
- Stable source IDs are HeavyM media IDs.
- Playlist children are not top-level sources. A playlist child created by this API returns `index 0` plus playlist identity fields.

Supported Source Types In V1

`/sources/add` supports:

- `VIDEO`
- `IMAGE`
- `GIF`
- `NDI`
- `PLAYLIST`

Text, DeckLink, live camera, shader, and Spout/Syphon source creation are out of scope for this first source-management version.

Add Source

Send `/sources/add`.

Minimum arguments:

```
/sources/add "<TYPE>" "<SOURCE_PATH_OR_NAME>"
```

Examples:

```
/sources/add "VIDEO" "D:/media/intro.mp4"  
/sources/add "IMAGE" "D:/media/logo.png" "Show/Logos"  
/sources/add "GIF" "D:/media/countdown.gif" "duration=12000" "loop=1"
```

```
/sources/add "NDI" "Main Stage"  
/sources/add "PLAYLIST" "Intro Playlist"
```

For file-backed sources, the second argument is an existing file path. The file extension must match the requested type.

For **NDI**, the second argument is the NDI sender/stream name.

For **PLAYLIST**, the second argument is the new playlist name.

Add Source Destination

The optional third positional argument is a destination. It can name either:

- A source-bin folder path, for example **Show/Intro**.
- An existing playlist name or playlist ID, for adding the new source inside the playlist.

Examples:

```
/sources/add "VIDEO" "D:/media/intro.mp4" "Show/Intro"  
/sources/add "VIDEO" "D:/media/intro.mp4" "Intro Playlist"  
/sources/add "VIDEO" "D:/media/intro.mp4" "42"
```

If the destination matches an existing playlist, the source is inserted in that playlist. Otherwise HeavyM treats it as a folder path.

If the folder path does not exist, HeavyM adds the new source at the source-bin root. The success reply returns the actual **folderpath**.

Explicit destination keys are also accepted:

```
/sources/add "VIDEO" "D:/media/intro.mp4" "folder=Show/Intro"  
/sources/add "VIDEO" "D:/media/intro.mp4" "playlist=Intro Playlist"  
/sources/add "VIDEO" "D:/media/intro.mp4" "playlistid=42"
```

playlist and **playlistid** are strict. If the playlist does not exist, the command fails with **PlaylistNotFound**.

Add Source Optional Parameters

All settings are optional. When omitted, HeavyM uses the same defaults as a source created from the UI/import flow.

Accepted optional **key=value** arguments:

Key	Values	Applies to
name	Display name	All supported types
playbackmode or restartmode	normal, restart, skip, or enum number	All supported types
restartwithseq	0, 1, true, false, on, off	All supported types
seqid or sequenceid	Stable HeavyM sequence ID	All supported types
seqnumber or sequencenumber	One-based sequence position	All supported types
loop or loopmode	0, 1, loop, stop, freeze, follow	VIDEO, IMAGE, GIF, NDI
duration, durationms, or durationtime	Milliseconds	IMAGE, GIF, NDI
beatduration, durationbeat, or durationbeats	Beat count like 4 or 4/1	IMAGE, GIF, NDI
durationmode	time, ms, beats	IMAGE, GIF, NDI

Example with all common parameters:

```
/sources/add "VIDEO" "D:/media/intro.mp4" "loop=1" "duration=12000"
"restartwithseq=1" "seqnumber=3" "playbackmode=restart"
```

For python-osc, the same command can be sent as:

```
client.send_message(
    "/sources/add",
    [
        "VIDEO",
        "D:/media/intro.mp4",
        "loop=1",
        "duration=12000",
        "restartwithseq=1",
        "seqnumber=3",
        "playbackmode=restart",
    ],
)
```

Add Source Reply

Success reply example:

```
#bundle
/sources/add/ok 1
/sources/add/status "Ok"
/sources/add/errorcode 0
/sources/add/error ""
/sources/add/id 81
/sources/add/index 7
/sources/add/name "Intro Loop"
/sources/add/type "VIDEO"
/sources/add/folderpath "Show/Intro"
/sources/add/destination "Show/Intro"
```

If the source is added inside a playlist, the reply also includes:

```
/sources/add/playlistid 42
/sources/add/playlistname "Intro Playlist"
/sources/add/playlistindex 3
```

Delete Source

Delete by one-based top-level index:

```
/sources/3/delete
/sources/index/3/delete
/sources/delete 3
/sources/delete "index=3"
```

Delete by stable media ID:

```
/sources/id/81/delete
/sources/delete "id=81"
```

Deleting by ID can target a top-level source or a playlist child. Deleting a playlist deletes its children as part of the normal HeavyM source deletion behavior.

Success reply example:

```
#bundle
/sources/id/81/delete/ok 1
/sources/id/81/delete/status "Ok"
/sources/id/81/delete/errorcode 0
/sources/id/81/delete/error ""
```

```
/sources/id/81/delete/id 81
/sources/id/81/delete/index 7
/sources/id/81/delete/name "Intro Loop"
/sources/id/81/delete/type "VIDEO"
/sources/id/81/delete/folderpath "Show/Intro"
```

Error Replies

Errors include both a numeric code and a descriptive string.

Example:

```
#bundle
/sources/add/ok 0
/sources/add/status "Error"
/sources/add/errordcode 1004
/sources/add/error "File path does not exist"
```

Invalid source type example:

```
# Request
/sources/add "IMAG" "D:/media/logo.png" "Show/Logos"

# Reply bundle
/sources/add/ok 0
/sources/add/status "Error"
/sources/add/errordcode 1002
/sources/add/error "Unsupported source type 'IMAG'. Supported types: VIDEO,
IMAGE, GIF, NDI, PLAYLIST"
```

Proposed error codes:

Code	Name	Meaning
0	Ok	Command succeeded
1000	InvalidAddress	OSC address is malformed or unsupported
1001	MissingArgument	A required argument is missing
1002	UnsupportedType	Source type is unknown or not implemented in v1
1003	InvalidArgument	Argument value is malformed or inconsistent
1004	InvalidPath	File path is missing or does not exist
1005	UnsupportedFileType	File extension does not match the requested source type
1006	SourceNotFound	Source ID or index does not resolve to a source

Code	Name	Meaning
1007	PlaylistNotFound	Explicit playlist destination does not resolve
1008	LicenseDenied	Requested source type is blocked by the current license
1009	CreationFailed	HeavyM could not create the source after parsing succeeded
1010	DeleteFailed	HeavyM could not delete the source after resolving it
1011	UnsupportedOperation	Operation is recognized but not supported in this version