AD-DA.ca, Open Sound Control (OSC)

Plugin Description

The AD-DA.ca OSC plugin adds Open Sound Control send and receive workflows to Q-SYS designs over UDP. It can build reusable OSC command libraries, trigger commands from control pin activity, receive and parse incoming OSC messages, and optionally bridge those messages into Q-SYS Named Components and Named Controls.

This plugin is intended for systems that need flexible OSC transport between Q-SYS and external devices or software such as QLab, Yamaha consoles, lighting systems, and custom controllers.

Key capabilities:

- Send OSC commands from manual triggers or input events.
- Receive and filter OSC messages by address pattern.
- Reuse command libraries through import/export text data.
- Substitute input values into OSC addresses and arguments with placeholders such as `{1}`.
- Forward incoming UDP packets to follower instances or external destinations.
- Use internal OSC libraries to address Q-SYS Named Components and Named Controls directly.

Important notes:

- UDP only. TCP is not supported.
- OSC bundles are not supported.
- When Send mode binds a local UDP port, that sender instance becomes the effective leader for that port and can relay traffic to follower-style listeners through notifications.

Configuration Overview

Plugin version

This document is written for the plugin:

- AD-DA.ca OSC V1.3.1.0

QDS compatibility

This plugin has been developed and tested on:

- Q-SYS Designer version 9.10.2
- Q-SYS Designer version 10.0.X (Emulation Mode supported)

Release note

2025-11 : 1.3.1.0

- Updated UI layout and help screenshots.
- Added debounce and input event handling controls in Send mode.
- Added UDP bind port support and packet forwarding options.
- Added Named Components and Named Controls receive libraries.
- Added multicast support in Receive mode.
- Added Yamaha-specific argument conversion types.

2025-10-03 : 1.3.0.0

- Changed NIC Selection (Receive Mode) from ALL to ANY
- Fixed network compatibility issues for QDS 10.0.X Emulation Mode
- Resolved combobox and listbox integer text field issues for QDS 10.0.X
- Added support for new license types
- Added support for Boolean type arguments (true and false)
- New argument type "INPUT TO BOOLEAN" - converts "false" or 0 to boolean false, converts "true" or 1 to boolean true. Useful for converting input {1} to T/F, like a mute button. Accepts as true: true, "TRUE", "true", 1, 1.0, "1", "1.0"
- Added unique UUID to newly created command names to ensure unique names
- Fixed UI glitches in the command's library page
- Enhanced Internal Components and Named Controls Library - now supports multiple argument types:
    - If arg type is string, modifies the .String of the control
    - If arg type is boolean (true or false), changes the .Boolean of the control
    - If arg type is FLOAT, affects the .Position of the control
    - If arg type is INT32, affects the .Value of the control
    - If arg type is string and the string is "Trigger", triggers the control
- Enhanced Command Library functionality - clearing the field now deletes/resets the command library. Entering "TEMPLATE" loads a library with examples (loaded by default on first plugin use)
- Added ability to bind UDP port when sending messages to ensure destination can reply on a known port
- Send Mode now sends via Notification incoming messages when BIND PORT is enabled
- Added message forwarding functionality with Forward to IP Address and Forward Port controls
- Receiver in Follower mode now only receives messages from selected PORT
- Receiver in Follower mode no longer disables port controls
- Enhanced message logging - all incoming messages are now printed in the log for better troubleshooting

2024-09-25 : 1.2.0.0

- Initial release to Asset Manager
- Breaking changes: control's name has changed

- New UI / Pages
- Add Duplicate function in command library
- Add Test function in command library

2024-06-21 : 1.1.0.0

- Add OSC receive function
- Add the ability to control Named Components and Named Controls via OSC messages.
- Add the ability to import/export the command's library in Send mode.

2024-06-12 : 1.0.0.0

- Initial Release

Properties

License

Type: `String`
Default value: `Empty`
Description: License key activated for the main core.

Licenses are issued on a "per core" basis, as in each core needs its own license to run as many instances of the plugin as needed on this particular core and licenses are not transferable from core to core.

License keys can be added while in Offline mode, in the Properties pane on the right. To purchase licenses, go to https://ad-da.ca.

The plugin can be emulated for free, allowing you to prepare everything without requiring you to have access to a core.

License Backup

Type: `String`
Default value: `Empty`
Description: License key activated for the backup core.

If your main Q-SYS Core fails, the backup Core automatically takes over without needing a license to function as a temporary replacement. However, if the design boots from the backup Core, a valid license is required. Additionally, if the backup Core is restarted (due to power loss or other reasons), the plugin instances cannot initialize without a valid license.

Mode

Type: `Enum`
Choices: `Send`, `Receive`
Default value: `Send`

Description: Selects whether the instance builds and transmits OSC commands or listens for and parses incoming OSC traffic.

Send Count

Type: `Integer`
Range: `4` to `32`
Default value: `4`
Description: Number of `Send [x]` command slots exposed in Send mode.

Input Count

Type: `Integer`
Range: `4` to `32`
Default value: `4`
Description: Number of `Input [x]` rows exposed in Send mode.

Receiver Mode

Type: `Enum`
Choices: `Leader`, `Follower`
Default value: `Leader`
Description: Defines how a Receive mode instance gets UDP packets when multiple plugin instances need the same source port.

Notes:

- `Leader` opens the UDP port directly and distributes packets locally to followers.
- `Follower` does not bind the UDP socket. It subscribes to packets already received by a leader on the selected port.
- Use `Follower` only when another OSC instance is already acting as leader for the same incoming port.

OSC Address Count

Type: `Integer`
Range: `4` to `32`
Default value: `4`
Description: Number of OSC address match rows exposed in Receive mode.

Argument Count

Type: `Integer`
Range: `4` to `32`
Default value: `4`
Description: Number of argument outputs exposed for each receive address row.

Debug Print

Type: `Enum`
Choices: `Function Calls`
Default value: `Function Calls`
Description: Enables internal debug output intended for troubleshooting and development.

Controls

General

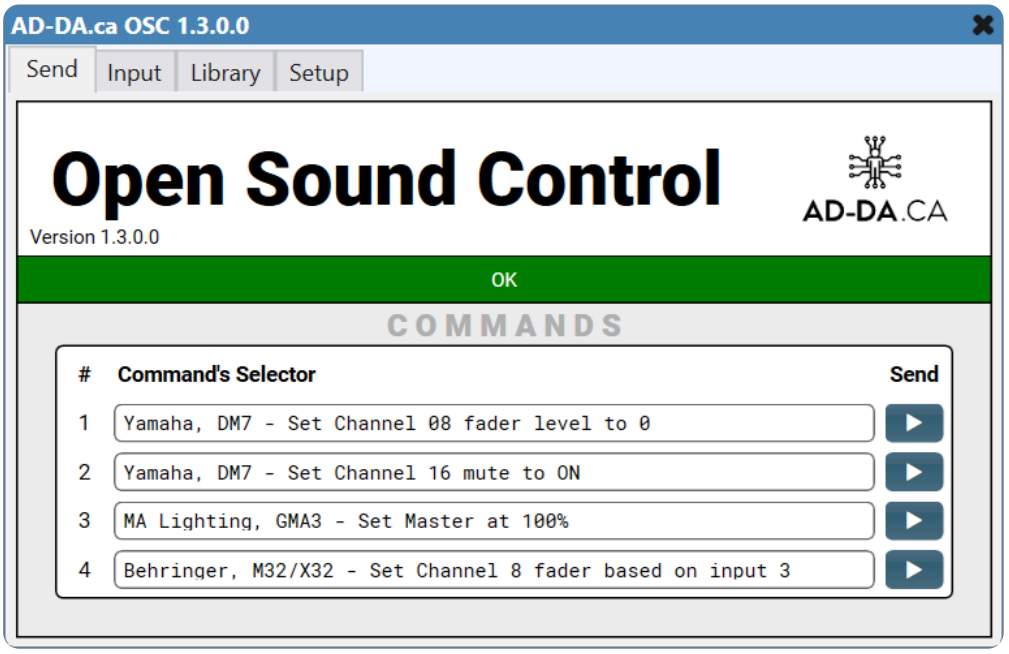
STATUS

Control name: `Status`
Type: `Indicator`
Controls Pin: `Output`
Description: Displays plugin runtime state.

Typical values:

- `OK`: Plugin is initialized and running.
- `FAULT`: Socket, network, or runtime error prevented operation.
- `MISSING`: License is missing or invalid.

Send Mode



SEND PAGE

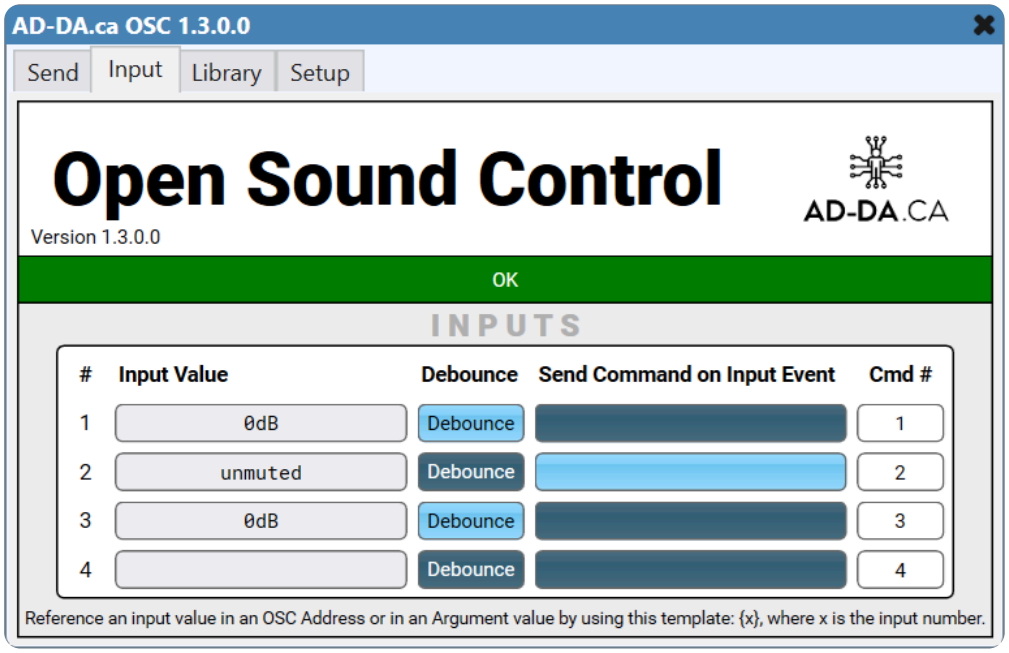
SendCommand [x]

Control name: `SendCommand [x]`
Type: `Text`
Controls Pin: `Input / Output`
Description: Assigns a command library entry to each send slot.

SendTrigger [x]

Control name: `SendTrigger [x]`
Type: `Button Trigger`
Controls Pin: `Input / Output`
Description: Sends the command assigned to `SendCommand [x]`.

INPUT PAGE



Input [x]

Control name: `Input [x]`
Type: `Indicator Text`
Controls Pin: `Input`
Description: Receives the raw input value used by command address and argument placeholders.

InputTrigger [x]

Control name: `InputTrigger [x]`
Type: `Button Toggle`
Controls Pin: `Input / Output`
Description: Enables or disables command transmission when `Input [x]` changes.

InputDebounce [x]

Control name: `InputDebounce [x]`
Type: `Button Toggle`

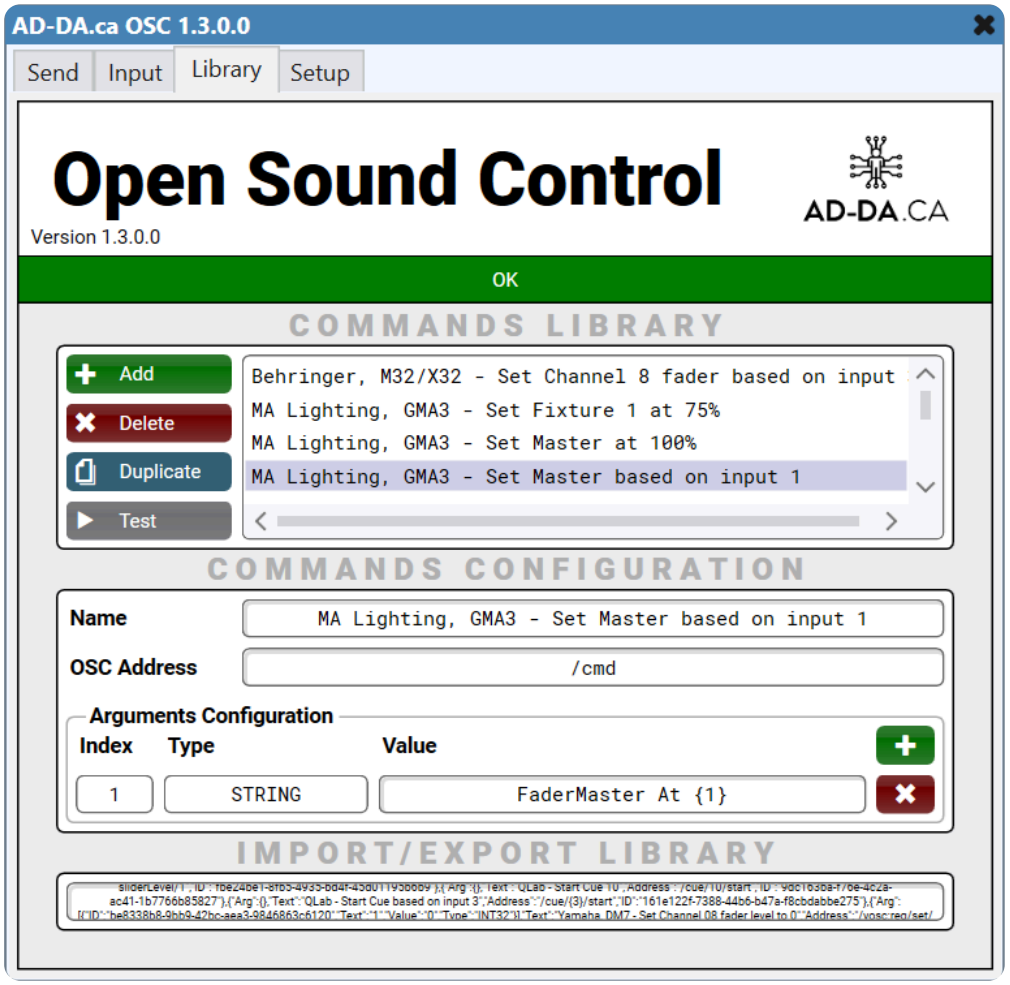
Controls Pin: `Input / Output`
Description: Filters rapid value changes on `Input [x]` to reduce unnecessary command bursts.

> This is useful when an input is driven by a knob or slider that produces many intermediate values.

InputTriggerButton [x]

Control name: `InputTriggerButton [x]`
Type: `Knob Integer`
Controls Pin: `Input / Output`
Range: `1` to `Send Count`
Description: Selects which send slot is triggered when the corresponding input event handler is enabled.

LIBRARY PAGE



CommandData

Control name: `CommandData`
Type: `Text`
Controls Pin: `Input / Output`

Default value: `*{ template library }*`
Description: Serialized command library field used for import, export, reset, and template loading.

> Pasting a saved library string loads the command library.
>
> Clearing the field resets the library.
>
> Entering `TEMPLATE` loads the bundled example library.

CommandList

Control name: `CommandList`
Type: `Text`
Controls Pin: `None`
Description: Selects the active command entry in the current command library.

CommandAdd

Control name: `CommandAdd`
Type: `Button Trigger`
Controls Pin: `None`
Description: Creates a new command entry with a unique identifier.

CommandDel

Control name: `CommandDel`
Type: `Button Trigger`
Controls Pin: `None`
Description: Deletes the currently selected command entry.

CommandDuplicate

Control name: `CommandDuplicate`
Type: `Button Trigger`
Controls Pin: `None`
Description: Creates a copy of the currently selected command entry.

CommandTest

Control name: `CommandTest`
Type: `Button Trigger`
Controls Pin: `None`
Description: Sends the currently selected command immediately for validation.

CommandName

Control name: `CommandName`
Type: `Text`
Controls Pin: `None`
Description: Human-readable name for the selected command.

CommandAddress

Control name: `CommandAddress`
Type: `Text`
Controls Pin: `None`
Description: OSC address to send for the selected command.

> Address placeholders such as `{1}` insert values from `Input [1]` at send time.

CommandArgList

Control name: `CommandArgList`
Type: `Text`
Controls Pin: `None`
Description: Selects which argument entry is being edited for the active command.

CommandArgAdd

Control name: `CommandArgAdd`
Type: `Button Trigger`
Controls Pin: `None`
Description: Adds a new argument row to the active command.

CommandArgDel

Control name: `CommandArgDel`
Type: `Button Trigger`
Controls Pin: `None`
Description: Deletes the selected argument row from the active command.

CommandArgType

Control name: `CommandArgType`
Type: `Text`
Controls Pin: `None`
Description: Sets the outgoing OSC argument type for the selected argument row.

Supported values:

- `INT32`
- `FLOAT`
- `STRING`
- `TRUE`
- `FALSE`
- `INPUT TO BOOLEAN`
- `INPUT TO YAMAHA GAIN`
- `MUTE TO BOOLEAN`
- `MUTE TO INT32`

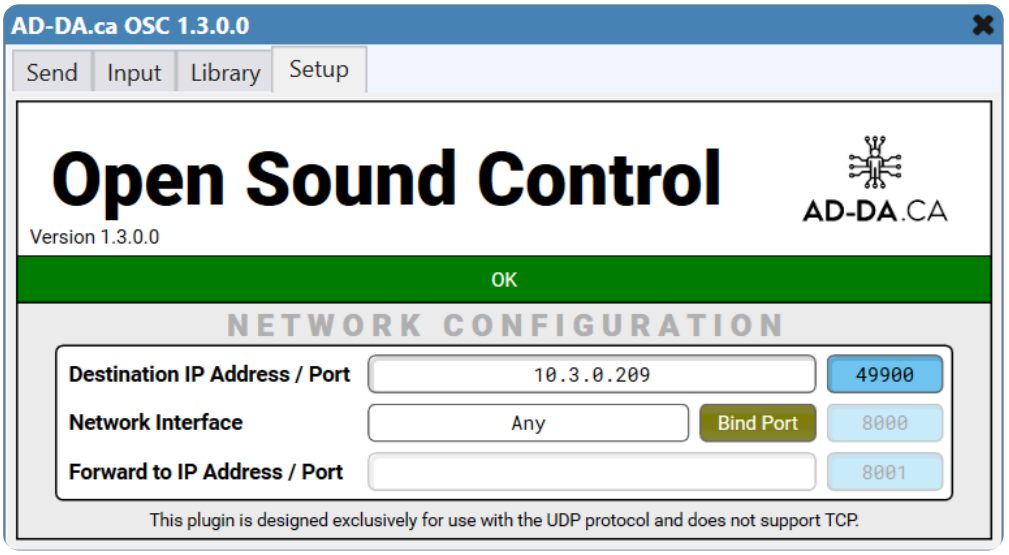
CommandArgValue

Control name: `CommandArgValue`
Type: `Text`
Controls Pin: `None`
Description: Value template for the selected argument row.

> Placeholders such as `{2}` insert values from `Input [2]`.

> `INPUT TO BOOLEAN` converts common string and numeric placeholder values into OSC boolean output.

SETUP PAGE



DestinationIP

Control name: `DestinationIP`
Type: `Text`
Controls Pin: `Input / Output`
Description: Target IP address for outgoing OSC messages.

> Multicast destination addresses automatically force bind-port behavior.

DestinationPort

Control name: `DestinationPort`
Type: `Knob Integer`
Controls Pin: `Input / Output`
Default value: `8000`
Range: `1` to `65535`
Description: Target UDP port for outgoing OSC messages.

Interface

Control name: `Interface`
Type: `Text`
Controls Pin: `Input / Output`
Description: Local network interface used for sending or for binding a reply port.

> `Any` allows the plugin to use the default route.

InterfacePort

Control name: `InterfacePort`
Type: `Knob Integer`
Controls Pin: `Input / Output`
Default value: `8000`
Range: `1` to `65535`
Description: Local UDP port to bind when reply handling or packet forwarding is required.

InterfacePortBind

Control name: `InterfacePortBind`
Type: `Button Toggle`
Controls Pin: `Input / Output`
Description: Enables local UDP port binding in Send mode.

> When enabled, the sender can receive replies on a known port.
>
> When enabled, incoming packets on that bound port are published to local followers and can also be forwarded externally.

ForwardIP

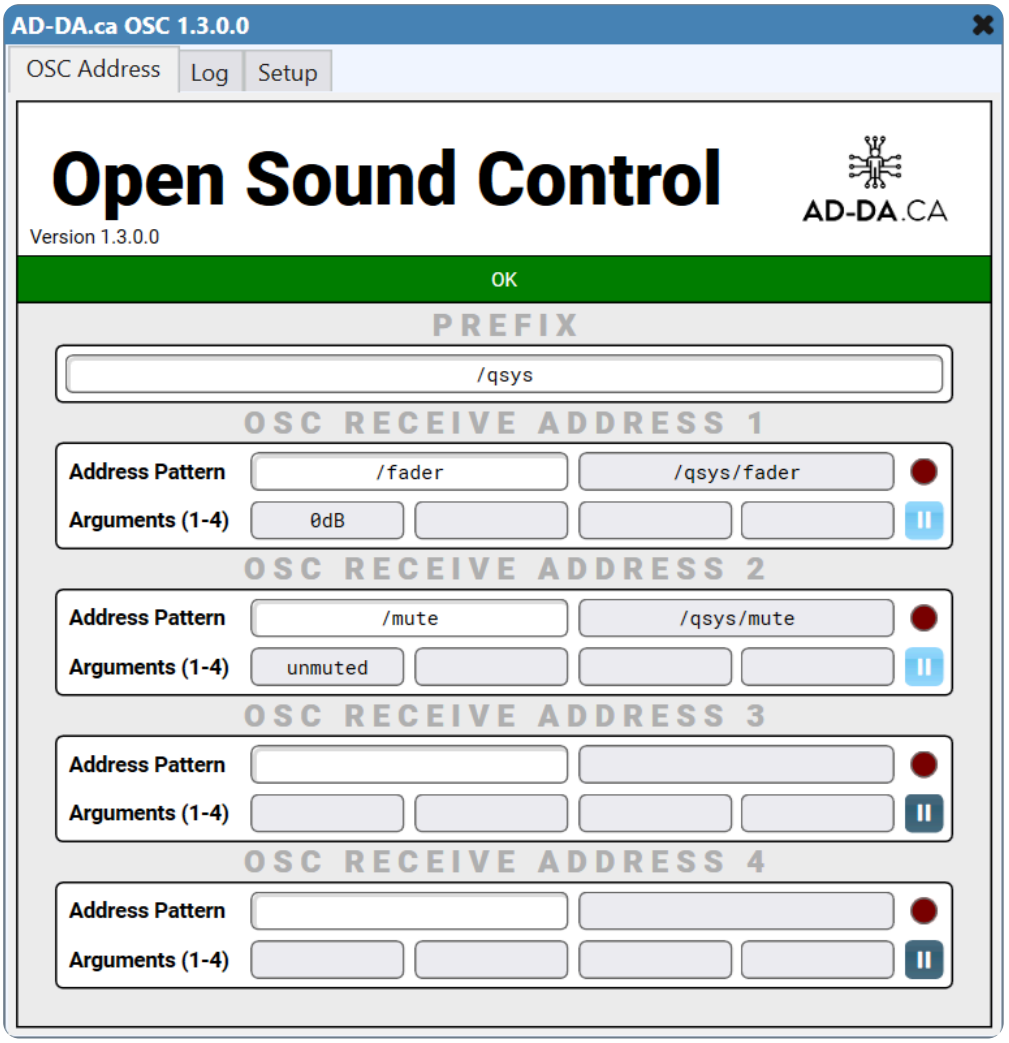
Control name: `ForwardIP`
Type: `Text`
Controls Pin: `Input / Output`
Description: Optional destination IP address for forwarding packets received by the bound sender socket.

ForwardPort

Control name: `ForwardPort`
Type: `Knob Integer`
Controls Pin: `Input / Output`
Default value: `8001`
Range: `1` to `65535`
Description: UDP port paired with `ForwardIP` for packet forwarding.

Receive Mode

OSC ADDRESS PAGE



Prefix

Control name: `Prefix`
Type: `Text`
Controls Pin: `Input / Output`
Description: Shared prefix prepended to every receive address row.

> This is useful when many messages share the same root path.

Address [x]

Control name: `Address [x]`
Type: `Text`
Controls Pin: `Input / Output`
Description: OSC address pattern to match for receive row `[x]`.

> The full pattern is `Prefix + Address [x]`.

> `*` wildcards can replace individual path segments.

AddressReceived [x]

Control name: `AddressReceived [x]`
Type: `Indicator Text`
Controls Pin: `Output`
Description: Displays the most recent OSC address received for row `[x]`.

AddressReceivedLed [x]

Control name: `AddressReceivedLed [x]`
Type: `Indicator LED`
Controls Pin: `Output`
Description: Pulses when row `[x]` matches an incoming OSC message.

AddressHold [x]

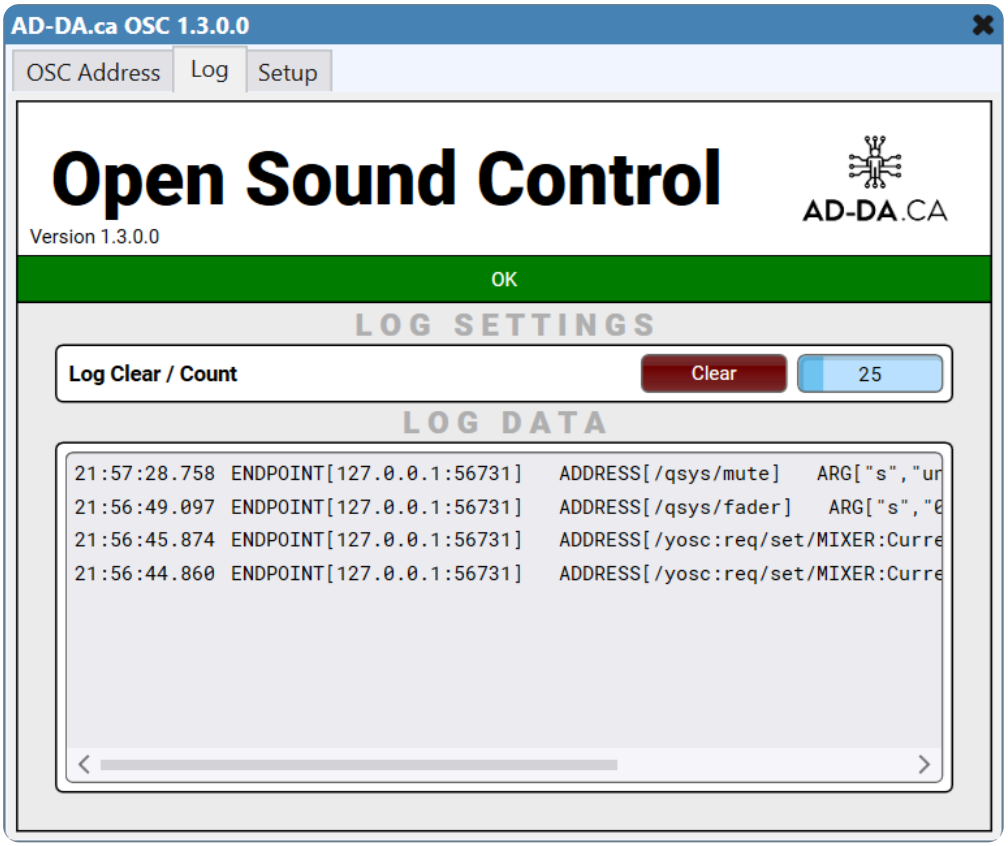
Control name: `AddressHold [x]`
Type: `Button Toggle`
Controls Pin: `Input / Output`
Description: Holds the last matched address and argument values for row `[x]`.

Address [x] Arg [y]

Control name: `Address [x] Arg [y]`
Type: `Indicator Text`
Controls Pin: `Output`
Description: Displays the parsed OSC argument `[y]` from the last message that matched row `[x]`.

> The number of available argument outputs per row is controlled by the `Argument Count` property.

LOG PAGE



Log

Control name: `Log`
Type: `Text`
Controls Pin: `Input / Output`
Description: Displays the receive log content.

> Incoming traffic is logged for troubleshooting, including messages outside the configured match list.

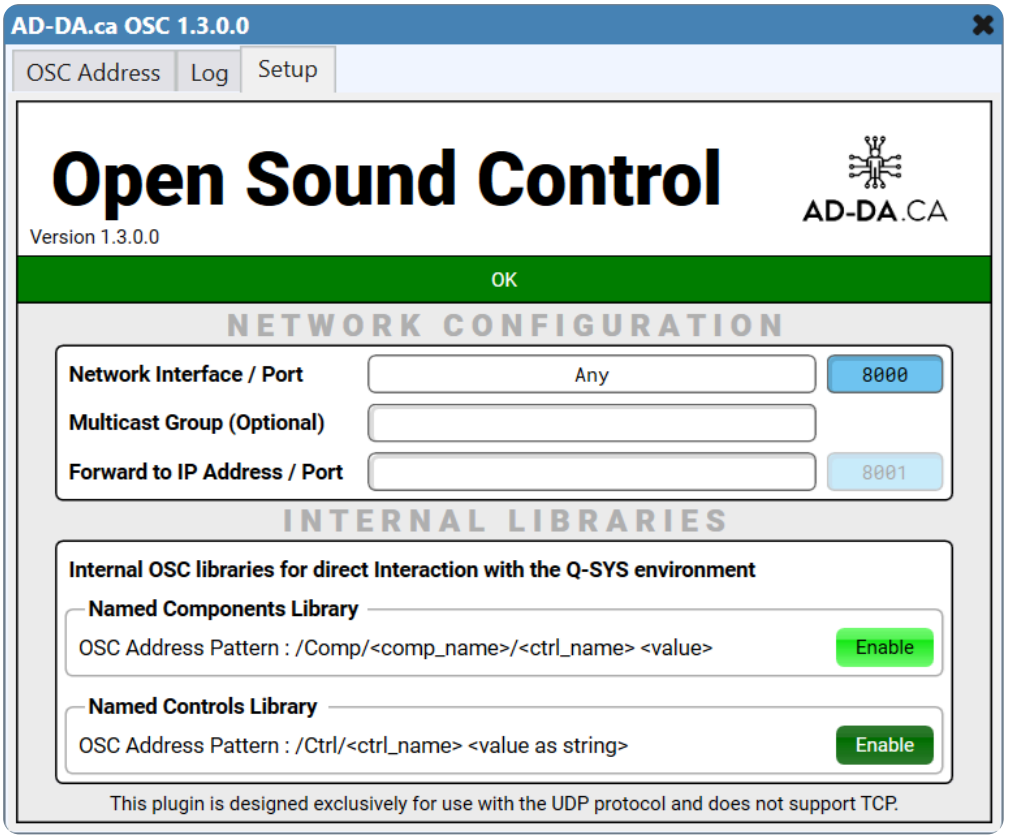
LogLength

Control name: `LogLength`
Type: `Knob Integer`
Controls Pin: `Input / Output`
Default value: `25`
Range: `10` to `100`
Description: Sets how many log lines are retained.

LogClear

Control name: `LogClear`
Type: `Button Trigger`
Controls Pin: `Input / Output`
Description: Clears the receive log immediately.

RECEIVE NETWORK SETUP



Interface

Control name: `Interface`
Type: `Text`
Controls Pin: `Input / Output`
Description: Local interface used by a leader receiver.

> In `Follower` mode this control is disabled because the instance does not bind the socket itself.

Port

Control name: `Port`
Type: `Knob Integer`
Controls Pin: `Input / Output`
Default value: `8000`
Range: `1` to `65535`
Description: UDP port used for listening or for follower subscription routing.

MulticastIP

Control name: `MulticastIP`
Type: `Text`
Controls Pin: `Input / Output`
Description: Optional multicast group joined by a leader receiver.

ForwardIP

Control name: `ForwardIP`
Type: `Text`
Controls Pin: `Input / Output`
Description: Optional destination IP address for forwarding incoming packets.

ForwardPort

Control name: `ForwardPort`
Type: `Knob Integer`
Controls Pin: `Input / Output`
Default value: `8001`
Range: `1` to `65535`
Description: UDP port paired with `ForwardIP` for packet forwarding.

CmdComp

Control name: `CmdComp`
Type: `Button Toggle`
Controls Pin: `Input / Output`
Description: Enables Named Components library handling for incoming OSC messages.

CmdCtrl

Control name: `CmdCtrl`
Type: `Button Toggle`
Controls Pin: `Input / Output`
Description: Enables Named Controls library handling for incoming OSC messages.

Send Behavior

Command library workflow

- Commands are stored as serialized JSON in `CommandData`.
- Each command contains one OSC address and zero or more typed arguments.
- `CommandTest` sends the selected command without needing an external trigger.
- New commands are created with unique identifiers so duplicates stay independent.

Placeholder substitution

- Address and argument values may reference send inputs with placeholders such as `{1}`, `{2}`, and `{3}`.
- Placeholder values are taken from the corresponding `Input [x]` control pin at the moment the command is sent.

Input-driven sending

- `InputTrigger [x]` enables automatic send behavior for that input row.
- `InputTriggerButton [x]` chooses which send slot is fired by that input row.
- `InputDebounce [x]` reduces excessive send events from fast-changing controls.

Outgoing argument types

- `INT32`, `FLOAT`, and `STRING` map to standard OSC types.
- `TRUE` and `FALSE` emit OSC boolean values.
- `INPUT TO BOOLEAN` converts common string and numeric values to OSC booleans.
- Yamaha-specific conversions are available for console workflows that need formatted or translated values.

Bind-port behavior

- When `InterfacePortBind` is off, the plugin sends from an automatically allocated socket.
- When `InterfacePortBind` is on, the plugin binds `InterfacePort` locally before sending.
- A bound sender can receive replies, publish those packets to followers on the same core, and optionally forward them to `ForwardIP:ForwardPort`.
- Multicast destinations force bind-port operation automatically.

Receive Behavior

Leader and follower operation

- `Leader` opens the local UDP port and receives packets directly.
- `Follower` subscribes to packets already received by a leader on the same configured port.
- This avoids multiple listeners competing for the same UDP port on the same core.

Address matching

- Each row listens to `Prefix + Address [x]`.
- `*` wildcards allow partial path matching.
- A successful match updates the received address text, LED, and argument outputs for that row.

Hold behavior

- With `AddressHold [x]` enabled, the most recent matched values remain available until a later valid match replaces them.

Logging and forwarding

- The log page captures incoming traffic for troubleshooting.
- A leader receiver can forward raw UDP packets to `ForwardIP:ForwardPort`.
- Followers receive locally published packets only for the configured `Port`.

Multicast

- If `MulticastIP` is populated, a leader receiver joins that multicast group after opening the local port.
- Followers do not join multicast directly.

Internal OSC Libraries

Named Components library

Enable `CmdComp` to let incoming OSC messages address Q-SYS Named Components directly.

Syntax: `/Comp/<comp_name>/<ctrl_name> <value_as_string>`

Rules:

- Spaces are not allowed in `<comp_name>`, `<ctrl_name>`, or `<value_as_string>`.
- If the control type resolves to string, the plugin writes `.String`.
- If the value resolves to boolean, the plugin writes `.Boolean`.
- If the value resolves to float, the plugin writes `.Position`.
- If the value resolves to integer, the plugin writes `.Value`.
- If the string value is `Trigger`, the target control is triggered.

Named Controls library

Enable `CmdCtrl` to let incoming OSC messages address Q-SYS Named Controls directly.

Syntax: `/Ctrl/<ctrl_name> <value_as_string>`

Rules:

- Spaces are not allowed in `<ctrl_name>` or `<value_as_string>`.
- The same type-handling rules used by the Named Components library also apply here.

Q-SYS naming notes

- In older Q-SYS Designer versions, rename the component block directly while offline.
- In Q-SYS Designer 9.5 and newer, use the component `Code Name` and ensure `Script Access` is set appropriately.
- Named Controls must be added to the Q-SYS Named Controls list before OSC messages can target them.

Support