AD-DA.ca, QueueSpeak

Plugin Description

QUEUESPEAK: ADVANCED Q-SYS CONFERENCING & VOTING PLUGIN

QueueSpeak transforms Q-SYS systems into a powerful conferencing and voting system, seamlessly integrating microphone, camera, and real-time polling management. Scalable from small meetings to large-scale venues, it leverages Q-SYS automation to streamline operations and provides post-event statistical exports for in-depth poll analysis.

BUILT FOR ANY ROOM, ANY BUDGET

QueueSpeak scales from a four-seat meeting room to a 260-seat assembly hall, and unlike closed conferencing systems, it is not tied to any single manufacturer: any microphone, any camera and any control surface can be combined in the same design. Build the most advanced, fully automated system the budget allows, or assemble a cost-effective, feature-rich room with only the pieces actually needed.

EFFORTLESS SETUP & CUSTOMIZATION

Get started quickly with pre-configured presets for common use cases or craft tailored setups by combining modes and options. Save and recall custom presets to suit any event, ensuring flexibility and ease of use.

DYNAMIC PRIORITIZATION & REQUEST LIST MANAGEMENT

Its multi-level priority system manages diverse participants with varying roles and privileges. The integrated Request List automates microphone and camera control, enabling smooth, hands-free operation for any event, small or large.

GRANULAR CONTROL, LIMITLESS POSSIBILITIES

Dedicated settings for each module put you in command. Fine-tune microphone, camera, and polling behaviors to deliver a polished, professional experience tailored to your audience and moderators.

Key Features

- Supports up to 260 seats / delegates with per-seat role, priority, audio, camera, and voting controls.
- Focus/Active List/Request List speaker-queue workflow with role profiles (Delegate, Chairman, Technician, Observer) and editable presets for common meeting formats.
- Fully customizable voting system with prepared polls, weighted/anonymous voting, role-aware eligibility, live moderator dashboards, real-time charts, and JSON/PDF report export.

- Automatic camera control: multiple camera positions per seat, Automatic Camera Preset Recall (ACPR), and audio-driven camera switching between active seats.
- Flexible audio routing (Main Mix, auxiliary group mixes, per-seat mix-minus and direct outputs) with audio-signal-driven request/activation/mute/camera-follow automation.
- Browser-based Web UCIs for every role: a lightweight Web control page for delegates (no physical hardware needed, part of Single Seat mode), and a full Dashboard Web UCI for moderators/administrators with role-based access control and an optional chrome-free Kiosk Mode.
- User Library mode: a delegate directory with PIN code and NFC tag recall, automatically restoring a seat's name, role, priority, gain, mixes, and vote weight when a delegate logs in.
- Fully localizable interface: a Language property drives translated vote reports and every connected web UI across 15 languages, with an optional per-browser Language override on the web control pages.

Use cases

- Parliament / Legislative Assemblies
- School Council / City Council Meetings
- United Nations / International Organizations
- Remote-led interactive classrooms
- Conference Panel Discussions with Audience Interaction
- Corporate Board Meetings
- Union Meetings
- Shareholders' General Assemblies

Configuration Overview

Plugin version

This document is written for the plugin:

- QueueSpeak V3.0.0.332

Release note

2026-08: V3.0.0.332

BREAKING CHANGE

This is a major update. Queue-facing labels and controls were renamed around Request List/Active List/Focus Seat terminology (Active Seat status is now Focus Seat status), and the Dashboard seat button action set was simplified from Set Active/Set Focus/Set Standby/Delete down to Add to Request List / Add to Active List.

If you previously used QueueSpeak V2, you should start from scratch with a new instance of this plugin rather than upgrading an existing design in place - the renamed controls/pins and reworked

workflow make migrating a V2 design impractical.

The focus of this release is turning QueueSpeak into a system that behaves like a traditional delegate microphone/conference system out of the box, while keeping the flexibility of an open Q-SYS design: any microphone/mixer, any camera (Q-SYS NC, Generic Visca, or selected Panasonic PTZ models), and any control surface (Q-SYS TSCG3, nnounce micnode2g, an iPad/tablet or any browser via the built-in Web UCI, or a fully custom UCI).

NEW FEATURE / IMPROVEMENT

- **New mode - Dashboard Web UCI:** a full moderator/administrator/guest dashboard delivered entirely through a browser.
- **New mode - User Library:** a standalone, JSON-backed delegate directory with PIN code and NFC tag recall.
- **New property - Presets JSON:** shows the preset JSON import/export controls, allowing all presets to be copied out of the plugin or pasted back in from JSON.
- **New property - Dashboard Seats Controls:** controls whether the dashboard creates per-seat buttons for direct moderator control.
- **New property - User Library:** shows the User Library recall controls in Single Seat mode, letting the seat connect to a User Library instance and recall a delegate's profile by PIN code or NFC tag.
- **New property - Internal Audio Component:** enables or disables the plugin's built-in audio processing and audio pins, for designs that handle audio processing entirely outside the plugin.
- **New property - Camera Count:** sets the number of camera slots created on the Components page.
- **New property - Camera Type:** selects the camera control method used for the configured camera slots - Q-SYS NC, Generic Visca, or a selected Panasonic PTZ model.
- **New property - Camera Visca Protocol:** selects the VISCA transport protocol used by Generic Visca cameras.
- **New property - Force Anonymous Voting:** forces all polls to behave as anonymous, regardless of each poll's own Vote Anonymous setting.
- **New property - Language:** drives translated vote reports and every connected web UI across 15 languages.
- **New Presets engine:** entire seat/vote/role configurations can now be saved, duplicated, and recalled as one-click presets, with editable default templates for common meeting workflows.
- **New Role engine:** every seat now has a Role (Delegate, Chairman, Technician, Observer) driving role-specific request/release actions, timer behavior, and vote eligibility, with a full Role Profile Editor.
- **New Timer page and controls:** a dedicated Timer with Play/Pause/Stop, elapsed and countdown display, and auto-start/timer-limit-action settings.
- **Standby camera router output:** the standby camera can now be assigned to its own dedicated output on the Camera/Mediacast router.

- **New About page:** licensing controls (License, License Backup) moved to a dedicated About page.
- **Complete new interface for the plugin:** redesigned Controls and Properties pages throughout every mode.
- **New Delegate Web UCI:** a lightweight, self-contained browser-based control page for delegates, part of Single Seat mode, requiring no physical hardware.
- **New Chairman Web UCI:** the Dashboard Web UCI mode, with role-based Web Users access (Guest, Operator, Administrator), Kiosk Mode, and an in-app Help popup with a guided tour.
- Much more.

2026-07: V2.1.0.0

BUG FIX

- Fixed seats becoming unconfigured when the Vote function is disabled.

NEW FEATURE / IMPROVEMENT

- Added "Internal Audio Component" property to let the user disable all internal audio components.
- Added support for the new Mediacast Router.
- Changed label to show support for the old Camera Router.
- Added SettingInterruptedSeat controls — ability to set the interrupted seat back to first in the list or cleared.

2026-03: V2.0.0.0

BUG FIX

- ADD automation behavior fixed (does not activate the seat except if it is the default seat).
- Disabling a seat now forces the muting of its microphone, even if it is locked open using the “Mute Lock” function.

NEW FEATURE

- Voting module
- Group mixes with audio output pins
- Seat mic lock & Group mix assignment
- Timers / countdown function
- Open microphones list to Dashboard
- Per-seat direct audio output pins (bypassing automatic gain)

IMPROVEMENT / MODIFICATION

- Increased maximum seat count to 260
- Redesigned Dashboard and Components GUI

2025-02: V1.0.0.0

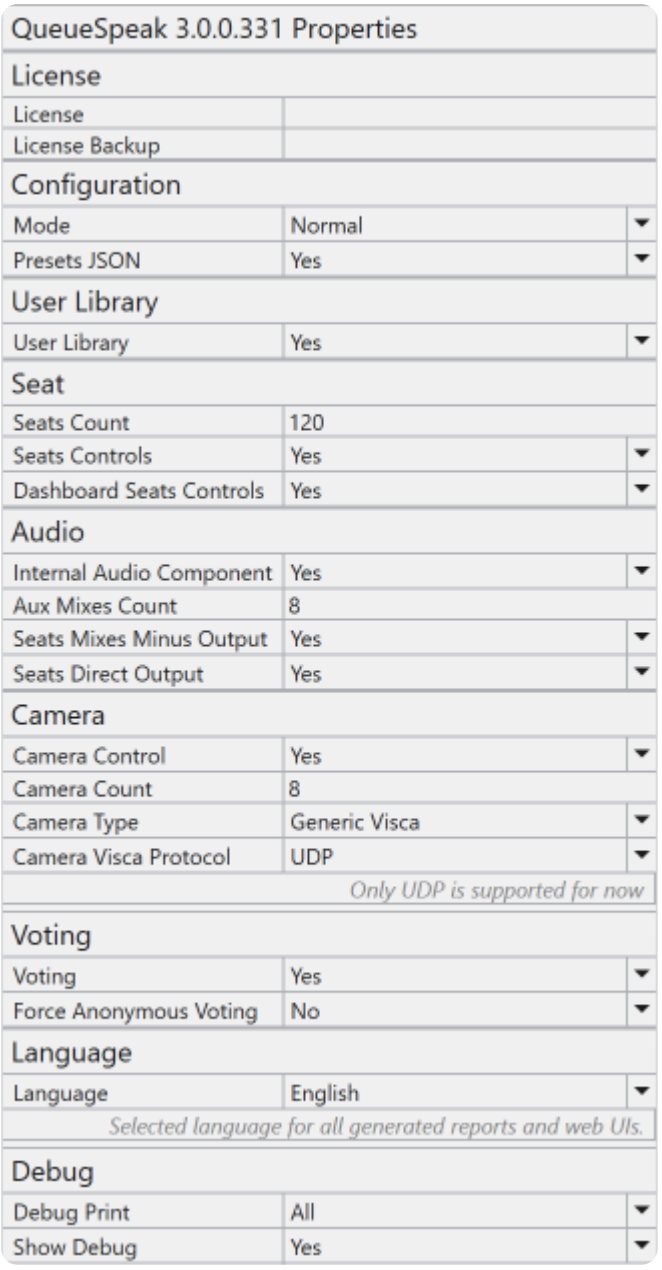
- Initial release to Asset Manager

QDS compatibility

This plugin has been developed and tested on:

- Q-SYS Designer version 9.13.x, 10.0.x, 10.1.x and 10.2.0

Properties



License

Type: `string`
Default Value: `Empty`
Description: License key used by the primary Q-SYS 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 used when the design runs on a backup Q-SYS 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.

Configuration

MODE

Type: `enum`
Choices: `Normal`, `Single Seat`, `User Library`, `Dashboard Web UCI`
Default Value: `Normal`
Description: Selects the operating mode for this plugin instance.

Use `Normal` for the main QueueSpeak instance. Use `Single Seat` to create a compact remote-control instance for one seat at a time. Single Seat mode is designed to connect to a Normal-mode instance and is useful when building custom UCIs, because its controls are generic instead of hard coded to one seat number. Use `User Library` to create a standalone, JSON-backed delegate directory instance that Single Seat instances can connect to for PIN/NFC-based delegate recall (see User Library Mode). Use `Dashboard Web UCI` to create a standalone browser-based moderator/administrator dashboard instance that connects to a Normal-mode instance (see Dashboard Web UCI Mode).

Important: When a plugin is used in Seat Mode, User Library Mode, or Dashboard Web UCI Mode, the Normal-mode instance must have its Script Access property set to `Script` or `All`. Single Seat and Dashboard Web UCI instances do not need Script Access enabled themselves to connect to a Normal-mode instance. A User Library instance, however, does need its own Script Access property set to `Script` or `All` whenever a Single Seat or Dashboard Web UCI instance connects to it for delegate recall/management (see User Library under Settings – Seat mode, and User Library under Settings – Dashboard Web UCI mode).

TSC TYPE

Type: `enum`
Choices: `None`, `Q-SYS TSCG3`, `nnounce micnode2g`, `Web`
Default Value: `Q-SYS TSCG3`
Description: Selects the touch screen controller integration available in `Single Seat` mode.

Use `None` when the Single Seat instance is controlled only by its exposed controls or a custom UCI. Use `Q-SYS TSCG3` to connect a Q-SYS touch screen controller and optional lightbar/sensor components. Use `nnounce micnode2g` to connect the micnode primary button and LED feedback. Use `Web` to serve a self-contained browser-based control page over an embedded web server, requiring no physical touch screen hardware (see Web Server under Settings – Seat mode).

This property is shown only when Mode is set to `Single Seat`.

USER LIBRARY

Type: `boolean`
Default Value: `false`
Description: Shows the User Library recall controls in `Single Seat` mode, letting the seat connect to a User Library instance and recall a delegate's profile by PIN code or NFC tag.

This property is visible in `Normal` and `Single Seat` modes (hidden in `User Library` and `Dashboard Web UCI` modes), but only has an effect in `Single Seat` mode. See User Library Mode for details on the standalone directory instance this connects to.

PRESETS JSON

Type: `boolean`
Default Value: `false`
Description: Shows the preset JSON import/export controls, allowing all presets to be copied out of the plugin or pasted back in from JSON.

Seat

SEATS COUNT

Type: `integer`
Min: `4`
Max: `260`
Default Value: `8`
Description: Sets the number of delegate seats created by the plugin, not including Seat 0 (Floor).

SEATS CONTROLS

Type: `boolean`
Default Value: `true`
Description: Shows or hides the seat configuration pages in the plugin interface.

Defines if the "Seats 1-x" and "Vote Seats" tabs are to be available in the plugin's interface.

Disabling Seat Controls is particularly useful in Seat Mode, where seat tabs become redundant and are automatically hidden.

DASHBOARD SEATS CONTROLS

Type: `boolean`
Default Value: `true`
Description: Controls whether the dashboard creates per-seat buttons for direct moderator control.

When enabled, the Dashboard page includes one button per configured seat. These buttons can be used for the selected dashboard action, Add to Request List or Add to Active List. Disable this property when dashboard seat buttons are not needed or when the seat count is high and the dashboard is controlled from external UCIs.

When `Dashboard Seats Controls` is disabled, the per-seat Dashboard and Vote Dashboard seat controls are not created. This includes `DashboardSeatAction`, `DashboardSeatName0`, `DashboardSeat0`, `DashboardSeatNameN`, `DashboardSeatN`, `DashboardSeatMuteN`, `DashboardSeatMuteLockN`, `VoteDashboardSeatNameN`, `VoteDashboardSeatValueN`, and `VoteDashboardSeatCastedN`.

Audio

INTERNAL AUDIO COMPONENT

Type: `boolean`
Default Value: `true`
Description: Enables the plugin's internal audio processing and audio pins.

When set to "Yes", the plugin includes its built-in audio processing components (gain-sharing mixer, gain blocks, etc.) and exposes all audio pins. When set to "No", all internal audio components are omitted, and no audio pins are added to the plugin. This is useful when audio processing is handled entirely outside the plugin.

AUX MIXES COUNT

Type: `integer`
Min: `0`
Max: `8`
Default Value: `8`
Description: Sets how many auxiliary group mix outputs are created.

Determines the number of auxiliary audio mix groups available for flexible audio routing. Auxiliary mixes are labeled A through H in the plugin GUI. Each auxiliary mix adds a corresponding audio output pin to the plugin.

SEATS MIXES MINUS OUTPUT

Type: `boolean`
Default Value: `false`
Description: Enables one mix-minus audio output per seat.

When this property is set to "Yes", the plugin provides a "Seat Mix Minus N" audio output pin for each seat. These outputs carry the Floor Mix (the plugin's main audio output) with the respective seat's microphone signal removed, ensuring an optimized audio stream that enhances clarity by minimizing feedback and echo issues. When this property is set to "No", the plugin will provide a single audio output (Floor Mix) that includes all unmuted microphones and the audio-in signal from the seat [000].

SEATS DIRECT OUTPUT

Type: `boolean`
Default Value: `false`
Description: Enables one direct unmanaged audio output per seat.

When this property is set to "Yes", the plugin provides a "Seat Direct N" audio output pin for each seat. These outputs carry the unmanaged audio signal from each seat, bypassing the automatic gain/volume leveling by tapping the signal directly after the mute/unmute stage.

Camera

CAMERA CONTROL

Type: `boolean`
Default Value: `true`
Description: Enables or disables camera control features.

Determines whether the plugin's "Camera" features will be used. If set to "No", camera controls are not created in the interface, and the plugin will operate using only audio features.

For system integrators: When "Camera Control" is enabled, certain conditions must be met for the plugin to function correctly. These conditions are:

- At least one Mediacast router must be in the design.
- At least one camera must be in the design.
- Mediacast and camera(s) must be "Named Components" (see How to rename a component for more details).
- The "Script Access" property of the Mediacast and cameras must be set to "Script" or "All".

CAMERA COUNT

Type: `integer`
Min: `2`
Max: `20`

Default Value: `8`
Description: Sets the number of camera slots created on the Components page.

CAMERA TYPE

Type: `enum`
Choices: `Q-SYS NC`, `Generic Visca`, `Manufacturer, Model` **Default Value:** `Q-SYS NC`
Description: Selects the camera control method used for the configured camera slots.

Use `Q-SYS NC` for Q-SYS NC Series cameras controlled through named Q-SYS components. Use `Generic Visca` for VISCA-over-IP PTZ cameras. Use one of the `Panasonic` choices for a supported Panasonic PTZ model. In Generic Visca and Panasonic camera modes, camera fields use IP addresses and seat camera positions use preset numbers instead of stored Q-SYS PTZ position strings.

This property is shown only when Camera Control is enabled.

CAMERA VISCA PROTOCOL

Type: `enum`
Choices: `UDP`
Default Value: `UDP`
Description: Selects the VISCA transport protocol used by Generic Visca cameras. Only UDP is currently supported.

This property is shown only when Camera Control is enabled and Camera Type is set to `Generic Visca`.

AUTO FRAMING SUPPORT

Type: `boolean`
Default Value: `false`
Description: Shows Auto Frame controls for Q-SYS NC cameras when auto framing is available in the design.

Defines if the "Auto Framing" function is supported by the system. When set to "Yes", an "Auto Frame" button is added to the seat camera parameters page. This button will enable the auto framing function of the camera (if available).

This property is shown only when Camera Control is enabled and Camera Type is set to `Q-SYS NC`.

Voting

VOTING

Type: `boolean`
Default Value: `true`
Description: Enables the voting module and its related pages, controls, and pins.

Enables the voting functionality. When set to "Yes", the Vote Dashboard, Vote Seats, and Vote Editor pages become available. When set to "No", these pages are hidden. The voting module is an optional feature tied to the license. To access it, a license that includes this module must be purchased.

FORCE ANONYMOUS VOTING

Type: `boolean`
Default Value: `false`
Description: Forces all polls to behave as anonymous, regardless of each poll's Vote Anonymous setting.

When enabled, individual seat vote values are hidden from the Vote Dashboard and sanitized from exported JSON/report data even if a poll was configured with Anonymous Vote disabled. Aggregate totals, percentages, and result summaries remain available.

Debug

DEBUG PRINT

Type: `enum`
Choices: `None`, `Camera`, `Audio`, `Voting`, `Seats`, `ACPR`, `UserLibrary`, `All`
Default Value: `None`
Description: Selects which debug messages are printed to the Q-SYS script log.

Use `None` during normal operation. Select a specific area, or `All`, when troubleshooting plugin behavior with Q-SYS Designer's script log.

Language

LANGUAGE

Type: `enum`
Choices: `English`, `Spanish`, `French`, `German`, `Portuguese (Brazil)`, `Italian`, `Polish`, `Dutch`, `Swedish`, `Turkish`, `Chinese (Simplified)`, `Japanese`, `Korean`, `Arabic`, `Ukrainian`
Default Value: `English`
Description: Sets the language used for generated vote reports and every connected web UI (Dashboard Web UCI, Single Seat, User Library).

Only set on the main plugin instance (`Mode` = `Normal`); connected mode components read this Property remotely and are not configured independently.

Controls

Status

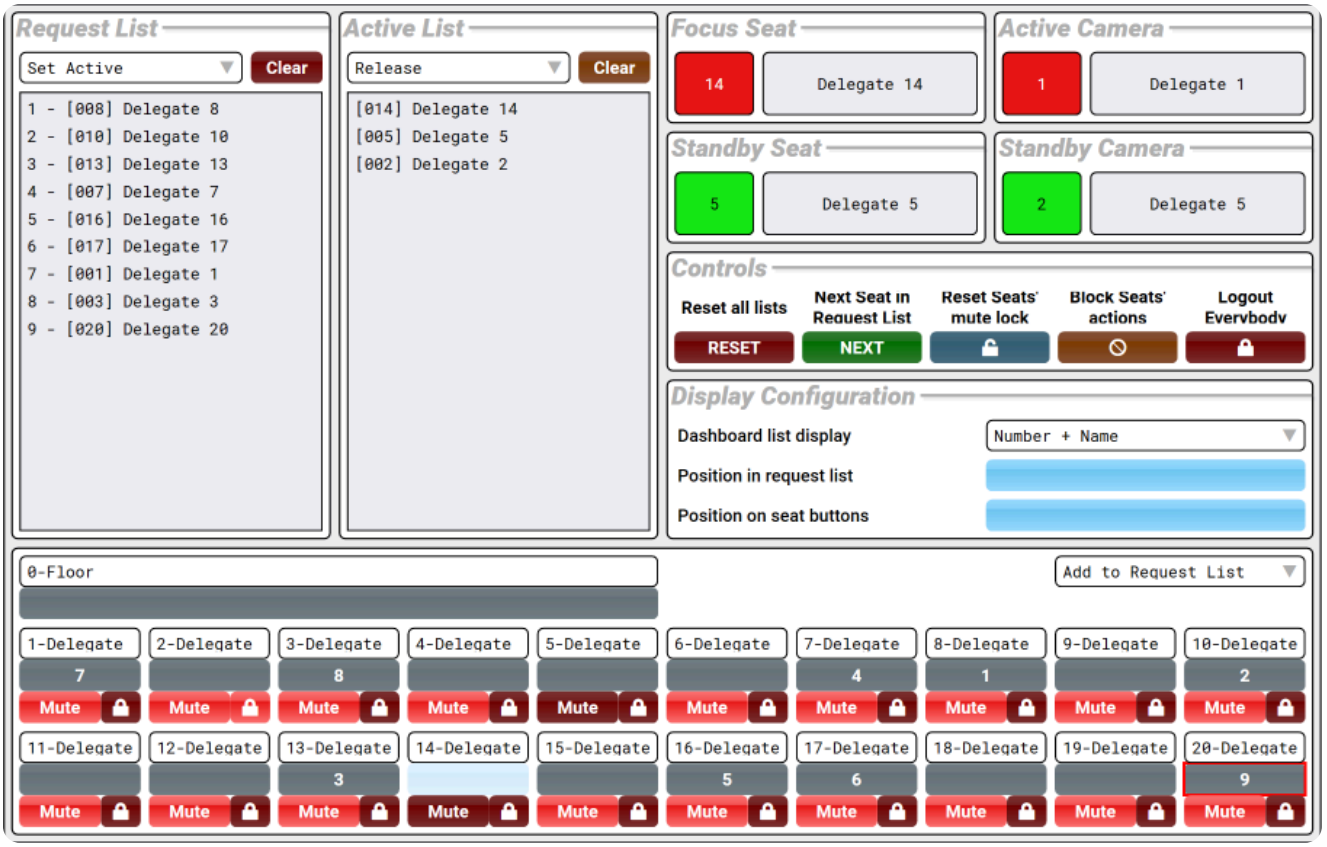
CONNECTION STATUS

Controls name: `Status`
Type: `Indicator-Status`
Pin Style: `Output`
Pin Name: `Status`
Description: Indicates verbose actual status of the plugin.

> OK – Plugin status (Active Seat or Voting Mode)

> Fault - [Message] (An error has occurred. Check the message details for further information)

Dashboard



DASHBOARD REQUEST LIST

Controls name: `DashboardRequestList`
Type: `Text-ListBox`
Pin Style: `Output`
Pin Name: `Dashboard~Request List~List`
Description: Displays the seats currently waiting in the Request List.

The list shows each waiting seat in request order. If Dashboard list display is enabled on the Dashboard tab, seats are displayed with both their number and name (e.g., [001] Delegate 1). The

list can also be used interactively with dashboard actions such as Set Focus, Set Active, Set Standby, and Delete.

DASHBOARD ACTIVE LIST

Controls name: `DashboardActiveList`
Type: `Text-ListBox`
Pin Style: `Input` / `Output`
Pin Name: `Dashboard~Active List~List`
Description: Displays seats currently in the Active List, including the Focus seat and any additional active/open seats.

If Dashboard list display is enabled on the Dashboard tab, each seat is shown with both its number and name (e.g., [001] Delegate 1). Otherwise, seats are displayed as "Seat-[###]".

DASHBOARD STATUS FOCUS SEAT

Controls name: `DashboardFocusSeat`
Type: `Indicator-Text`
Pin Style: `Output`
Pin Name: `Dashboard~Status~Focus Seat`
Description: Displays the numerical ID of the seat currently holding focus.

DASHBOARD STATUS FOCUS SEAT NAME

Controls name: `DashboardFocusSeatName`
Type: `Indicator-Text`
Pin Style: `Output`
Pin Name: `Dashboard~Status~Focus Seat Name`
Description: Displays the name of the seat currently holding focus.

DASHBOARD STATUS STANDBY SEAT

Controls name: `DashboardStandbySeat`
Type: `Indicator-Text`
Pin Style: `Output`
Pin Name: `Dashboard~Status~Standby Seat`
Description: Displays the ID of the next seat in line to receive focus, if any.

DASHBOARD STATUS STANDBY SEAT NAME

Controls name: `DashboardStandbySeatName`
Type: `Indicator-Text`
Pin Style: `Output`
Pin Name: `Dashboard~Status~Standby Seat Name`
Description: Displays the name of the next seat in line to receive focus, if any.

DASHBOARD STATUS ACTIVE CAMERA

Controls name: `DashboardActiveCamera`
Type: `Indicator-Text`
Conditional: `Camera Control` enabled.
Pin Style: `Output`
Pin Name: `Dashboard~Status~Active Camera`
Description: Displays the ID of the currently active camera. This ID corresponds to the camera's position in the "Components" tab.

DASHBOARD STATUS STANDBY CAMERA

Controls name: `DashboardStandbyCamera`
Type: `Indicator-Text`
Conditional: `Camera Control` enabled.
Pin Style: `Output`
Pin Name: `Dashboard~Status~Standby Camera`
Description: Displays the ID of the camera assigned to the standby seat. The system will always attempt to use a different camera than the active one, unless no other option is available. The standby camera is repositioned as soon as it is selected by the system, this occurs while the active camera is still in use, preventing camera movement delays between delegates.

DASHBOARD CONTROLS RESET [RST]

Controls name: `DashboardReset`
Type: `Button-Trigger`
Pin Style: `Input` / `Output`
Pin Name: `Dashboard~Controls~Reset`
Description: Clears the Request List and Active List, then returns focus to the Floor.

DASHBOARD CONTROLS NEXT [NEXT]

Controls name: `DashboardNext`
Type: `Button-Trigger`
Pin Style: `Input` / `Output`
Pin Name: `Dashboard~Controls~Next`
Description: Advances focus to the next seat. If the Request List contains seats, the first requested seat is promoted. If no seat is waiting, focus returns to the Floor. When Auto Promote is enabled and the Active List has room under Maximum Active List Length, Next instead tops up one seat from the Request List without releasing the current Focus - it only starts releasing/rotating the current Focus once the Active List is full.

DASHBOARD REQUEST LIST ACTION

Controls name: `DashboardRequestListAction`
Type: `Text-ComboBox`
Pin Style: `Input` / `Output`

Pin Name: `Dashboard~Request List~Action`
Description: Selects the action applied when a seat is selected from the Dashboard Request List.

Possible values:

- **Set Focus:** Moves the selected seat to the top of the Active List as the Focus seat. If the seat is not already active, it is added only when Maximum Active List Length has room, or when the list is full and Allow eviction when full is enabled. When eviction is needed, the last seat in the Active List is removed.
- **Set Active:** Adds the selected seat to the bottom of the Active List. If the Active List was empty, the seat becomes Focus. If the list is full, the Focus seat can be removed when Allow eviction when full is enabled; otherwise the action is blocked.
- **Set Standby:** Places the selected seat in the first position of the Request List, even when there is room in the Active List.
- **Delete:** Removes the selected seat from the Request List.

DASHBOARD REQUEST LIST CLEAR

Controls name: `DashboardClearRequestList`
Type: `Button-Trigger`
Pin Style: `Input` / `Output`
Pin Name: `Dashboard~Request List~Clear`
Description: Clears all seats from the Request List.

DASHBOARD ACTIVE LIST ACTION

Controls name: `DashboardActiveListAction`
Type: `Text-ComboBox`
Pin Style: `Input` / `Output`
Pin Name: `Dashboard~Active List~Action`
Description: Selects the action applied when a seat is selected from the Dashboard Active List.

Possible values:

- **Delete:** Removes the selected seat from the Active List.
- **Set Focus:** Moves the selected seat to the top of the Active List as the Focus seat. If the seat is not already active, it is added only when Maximum Active List Length has room, or when the list is full and Allow eviction when full is enabled. When eviction is needed, the last seat in the Active List is removed.

DASHBOARD ACTIVE LIST CLEAR

Controls name: `DashboardClearActiveList`
Type: `Button-Trigger`
Pin Style: `Input` / `Output`
Pin Name: `Dashboard~Active List~Clear`
Description: Clears all seats from the Active List and returns focus to the Floor.

DASHBOARD SEATS ACTION

Controls name: `DashboardSeatAction`
Type: `Text-ComboBox`
Conditional: `Dashboard Seats Controls` enabled.
Pin Style: `Input` / `Output`
Pin Name: `Dashboard~Seats~Action`
Description: Selects the action applied when a dashboard seat button is pressed.

Possible values:

- **Add to Request List:** Appends the selected seat to the bottom of the Request List.
- **Add to Active List:** Appends the selected seat to the bottom of the Active List. If the Active List was empty, the seat becomes Focus.

DASHBOARD CONTROLS RESET SEATS' MUTE LOCK

Controls name: `DashboardResetLock`
Type: `Button-Trigger`
Pin Style: `Input` / `Output`
Pin Name: `Dashboard~Controls~Reset Seat's Mute Lock`
Description: Disables any "Mute Lock" from the Dashboard or Seats.

DASHBOARD CONTROLS BLOCK

Controls name: `DashboardBlock`
Type: `Button-Toggle`
Pin Style: `Input` / `Output`
Pin Name: `Dashboard~Controls~Block`
Description: Disables the "Request" and "Release" buttons on seat interfaces.

DASHBOARD DISPLAY CONFIGURATION DASHBOARD LIST DISPLAY

Controls name: `DashboardListName`
Type: `Text-ComboBox`
Pin Style: `Input` / `Output`
Pin Name: `Dashboard~Display Configuration~Dashboard list display`
Description: When enabled, displays the seat name alongside the seat number in the Request List and Active List on the Dashboard. This setting is global and applies regardless of the active preset.

DASHBOARD DISPLAY CONFIGURATION POSITION IN REQUEST LIST

Controls name: `DashboardRequestListPosition`
Type: `Button-Toggle`
Pin Style: `Input` / `Output`
Pin Name: `Dashboard~Display Configuration~Position in request list`

Description: When enabled, each seat's Request List position field displays its current position in the Request List. This setting is global and applies regardless of the active preset.

DASHBOARD DISPLAY CONFIGURATION POSITION ON SEAT BUTTONS

Controls name: `DashboardShowPosition`
Type: `Button-Toggle`
Pin Style: `Input` / `Output`
Pin Name: `Dashboard~Display Configuration~Position on seat buttons`
Description: When enabled, displays the Request List position on the dashboard's seat buttons. This setting is global and applies regardless of the active preset.

DASHBOARD CONTROLS SEAT 0 NAME

Controls name: `DashboardSeatName0`
Type: `Indicator-Text`
Conditional: `Dashboard Seats Controls` enabled.
Pin Style: `Output`
Pin Name: `Dashboard~Seats~0~Name`
Description: Displays the assigned name for seat 0.

DASHBOARD FLOOR TRIGGER

Controls name: `DashboardSeat0`
Type: `Button-StateTrigger`
Conditional: `Dashboard Seats Controls` enabled.
Pin Style: `Input` / `Output`
Pin Name: `Dashboard~Seats~0~Activate`
Description: Manually activates Seat [000], and any automation associated with it (microphones, cameras). The button is gray when Seat [000] is inactive and turns light blue when active.

DASHBOARD SEAT N NAME

Controls name: `DashboardSeatNameN`
Type: `Indicator-Text`
Conditional: `Dashboard Seats Controls` enabled.
Pin Style: `Output`
Pin Name: `Dashboard~Seats~N~Name`
Description: Displays the assigned name for this seat.

DASHBOARD SEAT N TRIGGER

Controls name: `DashboardSeatN`
Type: `Button-StateTrigger`
Conditional: `Dashboard Seats Controls` enabled.
Pin Style: `Input` / `Output`
Pin Name: `Dashboard~Seats~N~Activate`

Description: Applies the selected Dashboard Seats Action to the corresponding seat. Each button label shows the seat number followed by its name. When Position on seat buttons is enabled on the Dashboard tab, buttons display their current Request List position when applicable.

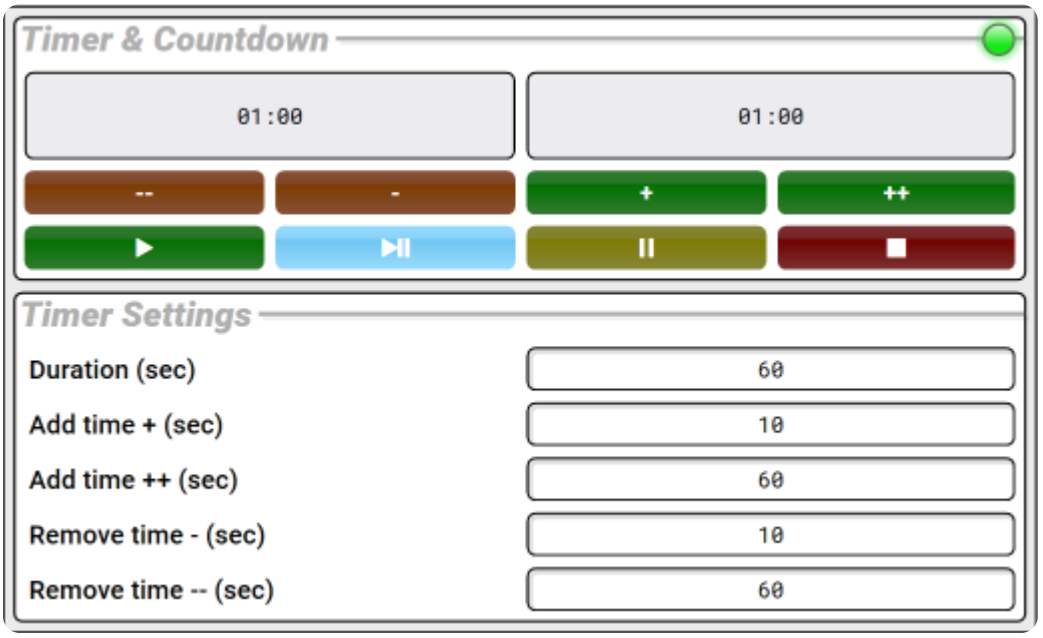
DASHBOARD SEAT N MUTE

Controls name: `DashboardSeatMuteN`
Type: `Button-Toggle`
Conditional: `Dashboard Seats Controls` enabled.
Pin Style: `Input` / `Output`
Pin Name: `Dashboard~Seats~N~Mute`
Description: Instantly mutes or unmutes the corresponding seat's microphone, regardless of its priority or active status. This allows the moderator to control any seat's microphone directly from the dashboard.

DASHBOARD SEAT N MUTE LOCK

Controls name: `DashboardSeatMuteLockN`
Type: `Button-Toggle`
Conditional: `Dashboard Seats Controls` enabled.
Pin Style: `Input` / `Output`
Pin Name: `Dashboard~Seats~N~Mute Lock`
Description: Locks the mute state in place, preventing built-in automations from controlling the seat. When activated, the seat remains in its current mute state until the moderator removes the lock.

Timer



A dedicated page for the elapsed/countdown timer shared with the Dashboard and Vote Dashboard status areas and the Dashboard Web UCI Timer card.

TIMER STATUS (LED)

Controls name: `TimerState`
Type: `Indicator-Led`
Pin Style: `Output`
Pin Name: `Timer~State~Led`
Description: Lights up while the timer is running.

TIMER

Controls name: `Timer`
Type: `Indicator-Text`
Pin Style: `Output`
Pin Name: `Timer~State~Timer`
Description: Displays the elapsed time since the timer started. If "Disabled" is selected for "Timer reached limit action" in the Settings tab, the timer will continue past the configured limit.

COUNTDOWN TIMER

Controls name: `Countdown`
Type: `Indicator-Text`
Pin Style: `Output`
Pin Name: `Timer~State~Countdown`
Description: Displays the remaining time until the countdown ends. If "Disabled" is selected for "Timer reached limit action" in the Settings tab, the countdown will continue into negative values after reaching zero.

TIMER CONTROLS REMOVE TIME (--)

Controls name: `TimerMinusMinus`
Type: `Button-Trigger`
Pin Style: `Input` / `Output`
Pin Name: `Timer~Controls~MinusMinus`
Description: Subtracts the amount set in the "Remove time -- (sec)" field from the countdown.

TIMER CONTROLS REMOVE TIME (-)

Controls name: `TimerMinus`
Type: `Button-Trigger`
Pin Style: `Input` / `Output`
Pin Name: `Timer~Controls~Minus`
Description: Subtracts the amount set in the "Remove time - (sec)" field from the countdown.

TIMER CONTROLS ADD TIME (+)

Controls name: `TimerPlus`
Type: `Button-Trigger`
Pin Style: `Input` / `Output`

Pin Name: `Timer~Controls~Plus`
Description: Adds the amount set in the "Add time + (sec)" field to the countdown.

TIMER CONTROLS ADD TIME (++)

Controls name: `TimerPlusPlus`
Type: `Button-Trigger`
Pin Style: `Input` / `Output`
Pin Name: `Timer~Controls~PlusPlus`
Description: Adds the amount set in the "Add time ++ (sec)" field to the countdown.

TIMER CONTROLS PLAY

Controls name: `TimerPlay`
Type: `Button-Trigger`
Pin Style: `Input` / `Output`
Pin Name: `Timer~Controls~Play`
Description: Starts the timer. Has no effect if the timer is already running.

TIMER CONTROLS PLAY PAUSE

Controls name: `TimerPlayPause`
Type: `Button-StateTrigger`
Pin Style: `Input` / `Output`
Pin Name: `Timer~Controls~PlayPause`
Description: Starts, pauses, and resumes the timer.

TIMER CONTROLS PAUSE

Controls name: `TimerPause`
Type: `Button-Trigger`
Pin Style: `Input` / `Output`
Pin Name: `Timer~Controls~Pause`
Description: Pauses the timer. Has no effect if the timer is not running.

TIMER CONTROLS STOP

Controls name: `TimerStop`
Type: `Button-Trigger`
Pin Style: `Input` / `Output`
Pin Name: `Timer~Controls~Stop`
Description: Stops and resets the timer.

TIMER SETTINGS DURATION

Controls name: `TimerDuration`
Type: `Text-Text Edit`
Pin Style: `Input` / `Output`

Pin Name: `Timer~Settings~Duration`
Description: Sets the timer duration. Values range from 10 seconds to 1 hour.

TIMER SETTINGS ADD TIME + (SEC)

Controls name: `TimerStepPlus`
Type: `Knob-Integer`
Pin Style: `Input` / `Output`
Pin Name: `Timer~Settings~StepPlus`
Description: Sets the amount of time added to the countdown by the "+" button. Values range from 1 to 600 seconds. (Default: 10)

TIMER SETTINGS ADD TIME ++ (SEC)

Controls name: `TimerStepPlusPlus`
Type: `Knob-Integer`
Pin Style: `Input` / `Output`
Pin Name: `Timer~Settings~StepPlusPlus`
Description: Sets the amount of time added to the countdown by the "++" button. Values range from 1 to 600 seconds. (Default: 60)

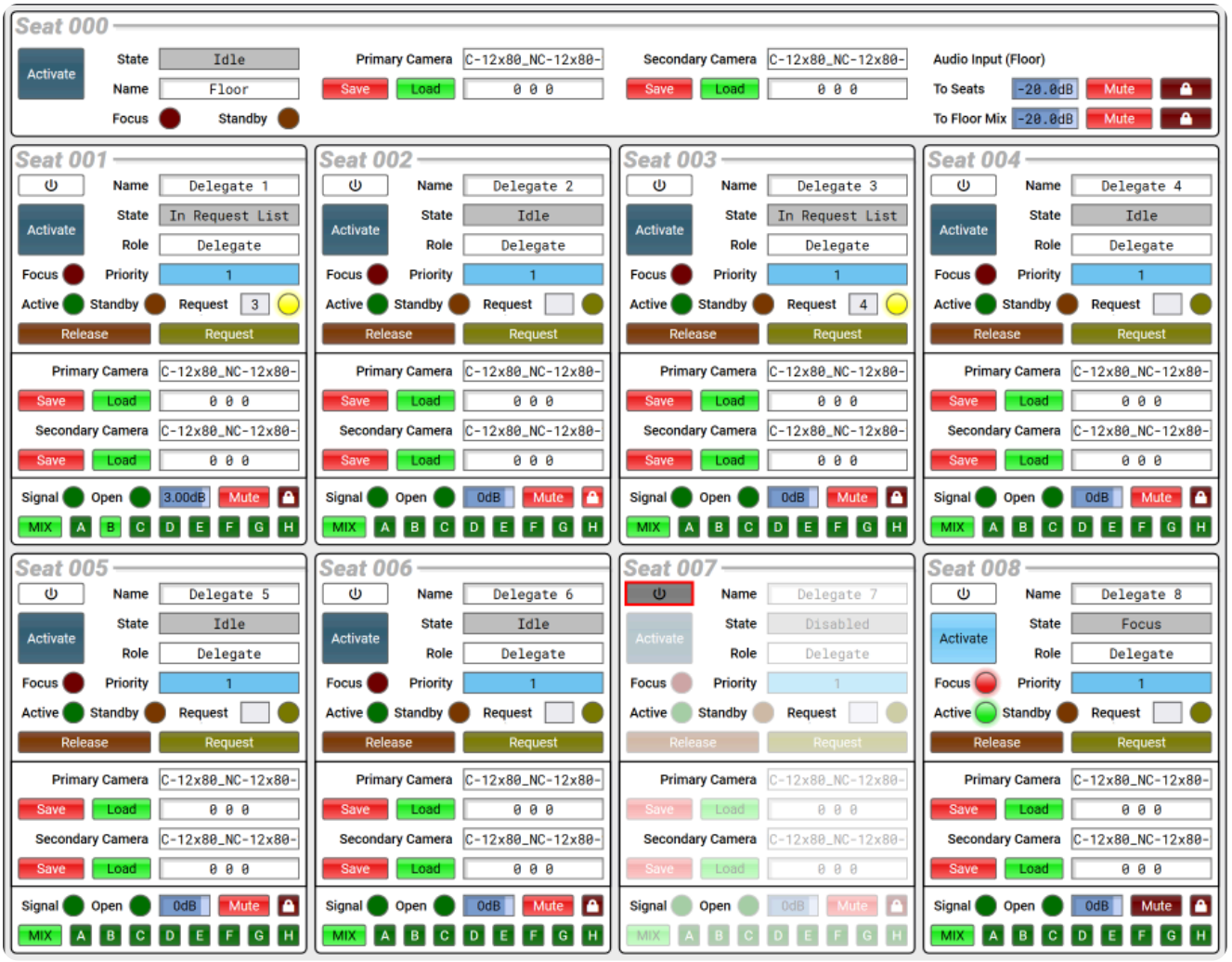
TIMER SETTINGS REMOVE TIME - (SEC)

Controls name: `TimerStepMinus`
Type: `Knob-Integer`
Pin Style: `Input` / `Output`
Pin Name: `Timer~Settings~StepMinus`
Description: Sets the amount of time subtracted from the countdown by the "-" button. Values range from 1 to 600 seconds. (Default: 10)

TIMER SETTINGS REMOVE TIME -- (SEC)

Controls name: `TimerStepMinusMinus`
Type: `Knob-Integer`
Pin Style: `Input` / `Output`
Pin Name: `Timer~Settings~StepMinusMinus`
Description: Sets the amount of time subtracted from the countdown by the "--" button. Values range from 1 to 600 seconds. (Default: 60)

Seats [1] - N



> Controls in this section are also available in Seat Mode.

SEATS 0 ACTIVATE

Controls name: `Seat0Activate`
Type: `Button-StateTrigger`
Pin Style: `Input` / `Output`
Pin Name: `Seats~0 (Floor)~Activate`
Description: Instantly activates Seat [000].

SEATS 0 STATE

Controls name: `Seat0State`
Type: `Indicator-Text`
Pin Style: `Output`
Pin Name: `Seats~0 (Floor)~Status~State`
Description: Shows the verbose state of the seat [000].

SEATS 0 NAME

Controls name: `Seat0Name`
Type: `Text-Text Edit`
Pin Style: `Input` / `Output`
Pin Name: `Seats~0 (Floor)~Config~Name`
Description: Set the name of seat [000]. (Default: “Floor”)

SEATS 0 FOCUS (LED)

Controls name: `Seat0Focus`
Type: `Indicator-Led`
Pin Style: `Output`
Pin Name: `Seats~0 (Floor)~Status~Focus`
Description: Lights up solid red when Seat [000] has focus.

SEATS 0 STANDBY (LED)

Controls name: `Seat0Standby`
Type: `Indicator-Led`
Pin Style: `Output`
Pin Name: `Seats~0 (Floor)~Status~Standby`
Description: Lights up solid orange when Seat [000] is the standby seat.

SEATS 0 PRIMARY CAMERA

Controls name: `Seat0PrimaryCamera`
Type: `Text-ComboBox`
Conditional: `Camera Control` enabled.
Pin Style: `Input` / `Output`
Pin Name: `Seats~0 (Floor)~Camera~Primary~Selector`
Description: Selects the primary camera from the cameras listed in the “Components" tab. At least one camera must be assigned to the seat [000] for the plugin to function properly.

SEATS 0 PRIMARY POSITION

Controls name: `Seat0PrimaryPosition`
Type: `Text-Text Edit`
Conditional: `Camera Control` enabled and `Camera Type` is `Q-SYS NC`.
Pin Style: `Input` / `Output`
Pin Name: `Seats~0 (Floor)~Camera~Primary~Position`
Description: Displays the stored PTZ preset for the primary camera position.

SEATS 0 PRIMARY SAVE

Controls name: `Seat0PrimaryPositionSave`
Type: `Button-Trigger`
Conditional: `Camera Control` enabled and `Camera Type` is `Q-SYS NC`.
Pin Style: `Input` / `Output`

Pin Name: `Seats~0 (Floor)~Camera~Primary~Save`
Description: Reads the Pan-Tilt-Zoom (PTZ) values from the primary camera, saves these values in the "Seats 0 Primary Position" field.

SEATS 0 PRIMARY LOAD

Controls name: `Seat0PrimaryPositionLoad`
Type: `Button-Trigger`
Conditional: `Camera Control` enabled and `Camera Type` is `Q-SYS NC`.
Pin Style: `Input` / `Output`
Pin Name: `Seats~0 (Floor)~Camera~Primary~Load`
Description: Sends the PTZ values stored in the "Seats 0 Primary Position" field to the primary camera, moving it to this position.

SEATS 0 PRIMARY AUTO FRAME

Controls name: `Seat0PrimaryAutoFraming`
Type: `Button-Toggle`
Conditional: `Camera Control` enabled, `Camera Type` is `Q-SYS NC`, and `Auto Framing Support` enabled.
Pin Style: `Input` / `Output`
Pin Name: `Seats~0 (Floor)~Camera~Primary~Auto Framing`
Description: Activates or deactivates the Auto Framing function of the assigned camera (if supported by the camera model). This button is only available when the “Auto Framing Support” property is set to “Yes”.

SEATS 0 PRIMARY PRESET

Controls name: `Seat0PrimaryPreset`
Type: `Text-Text Edit`
Conditional: `Camera Control` enabled and `Camera Type` is `Generic Visca` or a custom model. **Pin Style:** `Input` / `Output`
Pin Name: `Seats~0 (Floor)~Camera~Primary~Preset`
Description: Sets the Generic Visca preset number used for the primary camera position.

SEATS 0 SECONDARY CAMERA

Controls name: `Seat0SecondaryCamera`
Type: `Text-ComboBox`
Conditional: `Camera Control` enabled.
Pin Style: `Input` / `Output`
Pin Name: `Seats~0 (Floor)~Camera~Secondary~Selector`
Description: Selects the secondary camera from the cameras listed under the "Components" tab. At least one camera must be configured for the seat [000] for the plugin to function properly.

SEATS 0 SECONDARY POSITION

Controls name: `Seat0SecondaryPosition`
Type: `Text-Text Edit`
Conditional: `Camera Control` enabled and `Camera Type` is `Q-SYS NC`.
Pin Style: `Input` / `Output`
Pin Name: `Seats~0 (Floor)~Camera~Secondary~Position`
Description: Displays the Pan-Tilt-Zoom (PTZ) values for the secondary camera position.

SEATS 0 SECONDARY SAVE

Controls name: `Seat0SecondaryPositionSave`
Type: `Button-Trigger`
Conditional: `Camera Control` enabled and `Camera Type` is `Q-SYS NC`.
Pin Style: `Input` / `Output`
Pin Name: `Seats~0 (Floor)~Camera~Secondary~Save`
Description: Reads the Pan-Tilt-Zoom (PTZ) values from the secondary camera, saves these values in the "Seats 0 Secondary Position" field.

SEATS 0 SECONDARY LOAD

Controls name: `Seat0SecondaryPositionLoad`
Type: `Button-Trigger`
Conditional: `Camera Control` enabled and `Camera Type` is `Q-SYS NC`.
Pin Style: `Input` / `Output`
Pin Name: `Seats~0 (Floor)~Camera~Secondary~Load`
Description: Sends the PTZ values stored in the "Seats 0 Secondary Position" field to the secondary camera, moving it to the saved position.

SEATS 0 SECONDARY AUTO FRAME

Controls name: `Seat0SecondaryAutoFraming`
Type: `Button-Toggle`
Conditional: `Camera Control` enabled, `Camera Type` is `Q-SYS NC`, and `Auto Framing Support` enabled.
Pin Style: `Input` / `Output`
Pin Name: `Seats~0 (Floor)~Camera~Secondary~Auto Framing`
Description: Activates or deactivates the Auto Framing function of the assigned secondary camera when supported.

SEATS 0 SECONDARY PRESET

Controls name: `Seat0SecondaryPreset`
Type: `Text-Text Edit`
Conditional: `Camera Control` enabled and `Camera Type` is `Generic Visca` or a custom model. **Pin Style:** `Input` / `Output`
Pin Name: `Seats~0 (Floor)~Camera~Secondary~Preset`
Description: Sets the Generic Visca preset number used for the secondary camera position.

SEATS 0 TO SEATS GAIN

Controls name: `Seat0SeatGain`
Type: `Knob-Integer`
Conditional: `Internal Audio Component` enabled and `Seats Mixes Minus Output` enabled.
Pin Style: `Input` / `Output`
Pin Name: `Seats~0 (Floor)~Audio~To Seats Gain`
Description: Adjusts the Floor audio input level sent to the seats. (See Internal Audio Schematic for more information)

SEATS 0 TO SEATS MUTE

Controls name: `Seat0SeatMute`
Type: `Button-Toggle`
Conditional: `Internal Audio Component` enabled and `Seats Mixes Minus Output` enabled.
Pin Style: `Input` / `Output`
Pin Name: `Seats~0 (Floor)~Audio~To Seats Mute`
Description: Mutes the Floor audio input sent to the seats.

SEATS 0 TO SEATS MUTE LOCK

Controls name: `Seat0SeatMuteLock`
Type: `Button-Toggle`
Conditional: `Internal Audio Component` enabled and `Seats Mixes Minus Output` enabled.
Pin Style: `Input` / `Output`
Pin Name: `Seats~0 (Floor)~Audio~To Seats Mute Lock`
Description: Locks the "Seats 0 To Seats Mute" button in place. When this control is set to "Locked" (Enabled), it prevents any automation from making changes to the "Seat 0 To Seats Mute" button, preventing both unmuting and muting.

SEATS 0 TO MIX GAIN

Controls name: `Seat0MixGain`
Type: `Knob-Integer`
Conditional: `Internal Audio Component` enabled.
Pin Style: `Input` / `Output`
Pin Name: `Seats~0 (Floor)~Audio~To Mix Gain`
Description: Adjusts the Floor seat audio input level sent to the mix output.

SEATS 0 TO MIX MUTE

Controls name: `Seat0MixMute`
Type: `Button-Toggle`
Pin Style: `Input` / `Output`
Pin Name: `Seats~0 (Floor)~Audio~To Mix Mute`
Description: Mutes the Floor audio input from the mix output.

SEATS 0 TO MIX MUTE LOCK

Controls name: `Seat0MixMuteLock`
Type: `Button-Toggle`
Pin Style: `Input` / `Output`
Pin Name: `Seats~0 (Floor)~Audio~To Mix Mute Lock`
Description: Locks the "Seats 0 To Mix Mute" button in place. When this control is set to "Locked" (Enabled), it prevents any automation from making changes to the "Seats 0 To Mix Mute" button, preventing both unmuting and muting.

SEATS N ACTIVATE

Controls name: `SeatNActivate`
Type: `Button-StateTrigger`
Pin Style: `Input` / `Output`
Pin Name: `Seats~N~Activate`
Description: Instantly activates the seat.

SEATS N ENABLE

Controls name: `SeatNEnable`
Type: `Button-Toggle`
Pin Style: `Input` / `Output`
Pin Name: `Seats~N~Enable`
Description: Enables or disables a seat. When a seat is disabled, all its controls are deactivated and it is excluded from the voting system's total seat count. Seats are "Enabled" by default.

SEATS N NAME

Controls name: `SeatNName`
Type: `Text-Text Edit`
Pin Style: `Input` / `Output`
Pin Name: `Seats~N~Config~Name`
Description: Used to set a custom name for the seat. The default name is "Delegate N".

SEATS N ROLE

Controls name: `SeatNRole`
Type: `Text-ComboBox`
Pin Style: `Input` / `Output`
Pin Name: `Seats~N~Config~Role`
Description: Sets the seat role used by role-specific action profiles and vote eligibility.

Possible values:

- **Delegate:** Can vote and uses the Delegate role profile.
- **Chairman:** Can vote only when the poll's Chairman Can Vote option is enabled.
- **Technician:** Cannot vote; uses the Technician role profile.

- **Observer:** Cannot vote; uses the Observer role profile.

SEATS N STATE

Controls name: `SeatNState`
Type: `Indicator-Text`
Pin Style: `Output`
Pin Name: `Seats~N~Status~State`
Description: Shows the verbose state of the seat in the plugin's conferencing logic. This state represents the seat's operational role within the system's workflow, independent of individual controls such as microphone gain, mute, grouping or other per-seat parameters. The possible values are:

STATE	DESCRIPTION
`**Unconfigured**`	No camera has been assigned to this seat
`**Idle**`	The seat is enabled and available, but not currently requested or active
`**In Request List**`	The seat is waiting in the Request List
`**Active**`	The seat is in the Active List
`**Focus**`	The seat currently has focus; microphone and camera are positioned as configured
`**Override**`	The seat has exclusive focus during moderator Override; other seats are muted unless protected by mute lock
`**Voting**`	The seat is taking part in a vote
`**Disabled**`	The seat has been disabled via its Enable/Disable button

SEATS N PRIORITY

Controls name: `SeatNPriority`
Type: `Knob-Integer`
Pin Style: `Input` / `Output`
Pin Name: `Seats~N~Config~Priority`
Description: Numerical values from 1 to 5 for assigning a priority level to the seat. The higher the value, the higher the priority level. There are 3 different priority modes that determine how priority behaves within the plugin _See Active seat interruption (Priority) for more details.

SEATS N RELEASE

Controls name: `SeatNRelease`
Type: `Button-Trigger`
Pin Style: `Input` / `Output`
Pin Name: `Seats~N~Release`
Description: Triggers "Release" action as configured in the Settings tab.

_See, Release action (Idle), Release action (Active) and Release action (In Request List) for more details.

SEATS N REQUEST

Controls name: `SeatNRequest`
Type: `Button-Trigger`
Pin Style: `Input` / `Output`
Pin Name: `Seats~N~Request`
Description: Triggers "Request" action as configured in the Settings tab.

_See, Request action (Idle), Request action (Active) and Request action (In Request List) for more details.

SEATS N ACTIVE (LED)

Controls name: `SeatNActive`
Type: `Indicator-Led`
Pin Style: `Output`
Pin Name: `Seats~N~Status~Active`
Description: Lights up solid green when the seat is active.

For system integrators: In single-LED configurations, enable the Position in request list option on the Dashboard tab to expose request status without requiring a separate dashboard list.

SEATS N FOCUS (LED)

Controls name: `SeatNFocus`
Type: `Indicator-Led`
Pin Style: `Output`
Pin Name: `Seats~N~Status~Focus`
Description: Lights up solid red when the seat has focus.

SEATS N STANDBY (LED)

Controls name: `SeatNStandby`
Type: `Indicator-Led`
Pin Style: `Output`
Pin Name: `Seats~N~Status~Standby`
Description: Lights up solid orange when the seat is the standby seat.

SEATS N REQUEST LIST POSITION

Controls name: `SeatNRequestListPosition`
Type: `Indicator-Text`
Pin Style: `Output`
Pin Name: `Seats~N~Status~Request List Position`
Description: Displays the seat's current position in the Request List.

SEATS N IN REQUEST LIST (LED)

Controls name: `SeatNInRequestList`
Type: `Indicator-Led`
Pin Style: `Output`
Pin Name: `Seats~N~Status~In Request List`
Description: Lights up solid yellow when the seat is in the Request List.

SEATS N PRIMARY CAMERA

Controls name: `SeatNPrimaryCamera`
Type: `Text-ComboBox`
Conditional: `Camera Control` enabled.
Pin Style: `Input` / `Output`
Pin Name: `Seats~N~Camera~Primary~Selector`
Description: Selects this seat's primary camera from the cameras listed in the "Components" tab. Each seat must have at least one camera (primary or secondary) assigned for the plugin to function correctly.

SEATS N PRIMARY POSITION

Controls name: `SeatNPrimaryPosition`
Type: `Text-Text Edit`
Conditional: `Camera Control` enabled and `Camera Type` is `Q-SYS NC`.
Pin Style: `Input` / `Output`
Pin Name: `Seats~N~Camera~Primary~Position`
Description: Displays the stored Pan-Tilt-Zoom (PTZ) preset for the primary camera position.

SEATS N PRIMARY SAVE

Controls name: `SeatNPrimaryPositionSave`
Type: `Button-Trigger`
Conditional: `Camera Control` enabled and `Camera Type` is `Q-SYS NC`.
Pin Style: `Input` / `Output`
Pin Name: `Seats~N~Camera~Primary~Save`
Description: Reads the current PTZ values from the primary camera and saves them to the "Seats N Primary Position" field.

SEATS N PRIMARY LOAD

Controls name: `SeatNPrimaryPositionLoad`
Type: `Button-Trigger`
Conditional: `Camera Control` enabled and `Camera Type` is `Q-SYS NC`.
Pin Style: `Input` / `Output`
Pin Name: `Seats~N~Camera~Primary~Load`
Description: Sends the PTZ values stored in the "Seats N Primary Position" field to the primary camera, moving it to the saved position.

SEATS N PRIMARY AUTO FRAME

Controls name: `SeatNPrimaryAutoFraming`
Type: `Button-Toggle`
Conditional: `Camera Control` enabled, `Camera Type` is `Q-SYS NC`, and `Auto Framing Support` enabled.
Pin Style: `Input` / `Output`
Pin Name: `Seats~N~Camera~Primary~Auto Framing`
Description: Activates or deactivates the Auto Framing function of the assigned camera (if supported by the camera model). This button is only available when the “Auto Framing Support” property is set to “Yes”.

SEATS N PRIMARY PRESET

Controls name: `SeatNPrimaryPreset`
Type: `Text-Text Edit`
Conditional: `Camera Control` enabled and `Camera Type` is `Generic Visca` or a custom model. **Pin Style:** `Input` / `Output`
Pin Name: `Seats~N~Camera~Primary~Preset`
Description: Sets the Generic Visca preset number used for this seat's primary camera position.

SEATS N SECONDARY CAMERA

Controls name: `SeatNSecondaryCamera`
Type: `Text-ComboBox`
Conditional: `Camera Control` enabled.
Pin Style: `Input` / `Output`
Pin Name: `Seats~N~Camera~Secondary~Selector`
Description: Selects this seat’s secondary camera from the cameras listed in the "Components" tab. Each seat must have at least one camera (primary or secondary) assigned for the plugin to function correctly.

SEATS N SECONDARY POSITION

Controls name: `SeatNSecondaryPosition`
Type: `Text-Text Edit`
Conditional: `Camera Control` enabled and `Camera Type` is `Q-SYS NC`.
Pin Style: `Input` / `Output`

Pin Name: `Seats~N~Camera~Secondary~Position`
Description: Displays the stored Pan-Tilt-Zoom (PTZ) preset for the secondary camera position.

SEATS N SECONDARY SAVE

Controls name: `SeatNSecondaryPositionSave`
Type: `Button-Trigger`
Conditional: `Camera Control` enabled and `Camera Type` is `Q-SYS NC`.
Pin Style: `Input` / `Output`
Pin Name: `Seats~N~Camera~Secondary~Save`
Description: Reads the current PTZ values from the secondary camera and saves them to the "Seats N Secondary Position" field.

SEATS N SECONDARY LOAD

Controls name: `SeatNSecondaryPositionLoad`
Type: `Button-Trigger`
Conditional: `Camera Control` enabled and `Camera Type` is `Q-SYS NC`.
Pin Style: `Input` / `Output`
Pin Name: `Seats~N~Camera~Secondary~Load`
Description: Sends the PTZ values stored in the "Seats N Secondary Position" field to the secondary camera, moving it to the saved position.

SEATS N SECONDARY AUTO FRAME

Controls name: `SeatNSecondaryAutoFraming`
Type: `Button-Toggle`
Conditional: `Camera Control` enabled, `Camera Type` is `Q-SYS NC`, and `Auto Framing Support` enabled.
Pin Style: `Input` / `Output`
Pin Name: `Seats~N~Camera~Secondary~Auto Framing`
Description: Activates or deactivates the Auto Framing function of the assigned secondary camera when supported.

SEATS N SECONDARY PRESET

Controls name: `SeatNSecondaryPreset`
Type: `Text-Text Edit`
Conditional: `Camera Control` enabled and `Camera Type` is `Generic Visca` or a custom model. **Pin Style:** `Input` / `Output`
Pin Name: `Seats~N~Camera~Secondary~Preset`
Description: Sets the Generic Visca preset number used for this seat's secondary camera position.

SEATS N MIC SIGNAL (LED)

Controls name: `SeatNSignal`
Type: `Indicator-Led`
Conditional: `Internal Audio Component` enabled.
Pin Style: `Output`
Pin Name: `Seats~N~Audio~Mic Signal`
Description: Lights up solid green when the seat's microphone input exceeds the Microphone signal threshold (dB) level set in the Settings tab. The indicator stays lit until no audio signal is detected for a duration longer than the Microphone signal hold time (sec) setting, after which it turns off.

Important: This LED reflects mic signal detection, not whether the microphone is actively transmitting audio. The detection status is used by the plugin to trigger automations based on audio activity. The microphone may be open and transmitting even when this indicator is off, and vice versa. To check the actual mute state of the seat, refer to the "Seats N Mute" button. For more information, see the audio schematic section in the included Sample File.

SEATS N MIC OPEN (LED)

Controls name: `SeatNOpen`
Type: `Indicator-Led`
Conditional: `Internal Audio Component` enabled.
Pin Style: `Output`
Pin Name: `Seats~N~Audio~Mic Open`
Description: Reflects the "Input Open" status of the internal Gain-Sharing Automatic Mic Mixer. When lit, the seat's microphone signal level is sufficient for the Gain-Sharing Mixer to include it in the active mix path, the Floor Mix Output and/or the Seats Mix Outputs for any Groups the seat is assigned to.

SEATS N GAIN

Controls name: `SeatNGain`
Type: `Knob-Integer`
Conditional: `Internal Audio Component` enabled.
Pin Style: `Input` / `Output`
Pin Name: `Seats~N~Audio~Gain`
Description: Adjusts the priority gain of this seat's microphone in the Gain-Sharing Automatic Mic Mixer, affecting the Floor Mix and, when enabled, all Group mix outputs. This setting has no effect on the Seats Direct Output. For more information, see the audio schematic section in the included Sample File.

SEATS N MUTE

Controls name: `SeatNMute`
Type: `Button-Toggle`
Pin Style: `Input` / `Output`
Pin Name: `Seats~N~Audio~Mute`

Description: Mutes or unmutes the seat's microphone. When activated, the microphone is muted across all outputs, including Floor Mix, Seats Mix, and Seats Direct Output.

SEATS N MUTE LOCK

Controls name: `SeatNMuteLock`
Type: `Button-Toggle`
Pin Style: `Input` / `Output`
Pin Name: `Seats~N~Audio~Mute Lock`
Description: Locks the current mute state, preventing any automation from muting or unmuting this seat. Manual control via the “Seats N Mute” button remains available.

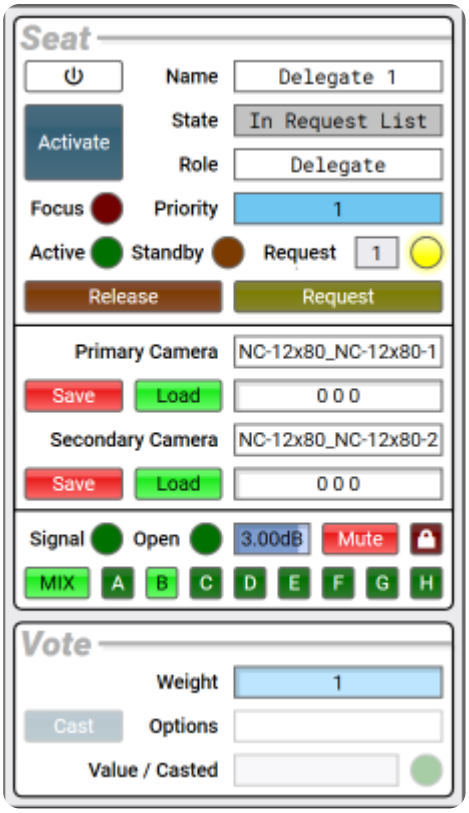
SEATS N MIX FLOOR MIX

Controls name: `SeatNFloorMix`
Type: `Button-Toggle`
Conditional: `Internal Audio Component` enabled and `Aux Mixes Count` is greater than `0`.
Pin Style: `Input` / `Output`
Pin Name: `Seats~N~Mix~Floor Mix`
Description: Routes the seat’s audio signal to the Floor Mix Output through the Gain-Sharing Automatic Mic Mixer. For more information, see the audio schematic section in the included Sample File.

SEATS N MIX GROUP [A-H]

Controls name: `SeatNGroupMix[A-H]`
Type: `Button-Toggle`
Conditional: `Internal Audio Component` enabled and `Aux Mixes Count` is greater than `0`.
Pin Style: `Input` / `Output`
Pin Name: `Seats~N~Mix~Group [A-H]`
Description: Routes the seat's audio signal to the Group [A-H] Seats Mix Output through the Gain-Sharing Automatic Mic Mixer. For more information, see the Internal Audio Schematic section in the included Sample File.

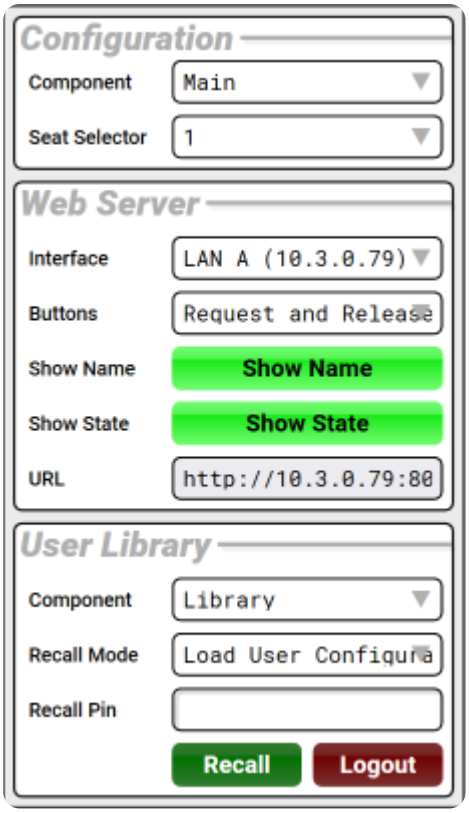
Seat Mode



The controls in Seat Mode are the same as the ones found in Seats 1-N, please refer to previous paragraphs for control description.

Seat Mode controls follow the same conditional rules as their Seats 1-N equivalents. Camera controls require `Camera Control` enabled, Q-SYS NC position controls require `Camera Type` set to `Q-SYS NC`, preset controls require `Camera Type` set to `Generic Visca` or a custom model, Auto Frame controls require `Auto Framing Support` enabled, audio-processing controls require `Internal Audio Component` enabled, and vote controls require `Voting` enabled.

Settings – Seat mode



COMPONENT

Controls name: `ConfigurationComponent`
Type: `Text-ComboBox`
Pin Style: `Input` / `Output`
Pin Name: `Configuration~Component`
Description: Displays available QueueSpeak plugins in "Default" mode; select the plugin to connect to for remote control. To appear in this list, the default plugins must have "Script access" set to "ALL" or "Script".

SEAT SELECTOR

Controls name: `ConfigurationSeatID`
Type: `Text-ComboBox`
Pin Style: `Input` / `Output`
Pin Name: `Configuration~Seat ID`
Description: Lists the seats available on the selected component. Selecting a seat links its controls to this instance for remote configuration.

TSC COMPONENT

Controls name: `ConfigurationTSCComponent`
Type: `Text-ComboBox`
Conditional: `Mode` is `Single Seat` and `TSC Type` is not `None`.
Pin Style: `Input` / `Output`
Pin Name: `Configuration~TSC Component`

Description: Selects the touch screen controller or micnode component used by this Single Seat instance.

For `Q-SYS TSCG3`, the selected component provides the primary button event. For `nnounce micnode2g`, the selected component provides the paging/PTT button and LED control feedback.

TSC LIGHTBAR COMPONENT

Controls name: `ConfigurationTSCLightbarComponent`
Type: `Text-ComboBox`
Conditional: `Mode` is `Single Seat` and `TSC Type` is `Q-SYS TSCG3`.
Pin Style: `Input` / `Output`
Pin Name: `Configuration~TSC Lightbar Component`
Description: Selects the Q-SYS TSCG3 lightbar component used for seat status feedback.

The lightbar turns green while the seat is unmuted. When muted, it reflects the current seat state: yellow for In Request List, red for Idle, green for Active, Focus, or Override, and off/black when no matching state applies.

TSC SENSORS COMPONENT

Controls name: `ConfigurationTSCSensorsComponent`
Type: `Text-ComboBox`
Conditional: `Mode` is `Single Seat` and `TSC Type` is `Q-SYS TSCG3`.
Pin Style: `Input` / `Output`
Pin Name: `Configuration~TSC Sensors Component`
Description: Selects the Q-SYS TSCG3 sensors component associated with the touch screen controller.

PRIMARY BUTTON ACTION

Controls name: `ConfigurationPrimaryButtonSelector`
Type: `Text-ComboBox`
Conditional: `Mode` is `Single Seat` and `TSC Type` is not `None`.
Pin Style: `Input` / `Output`
Pin Name: `Configuration~Primary Button`
Description: Selects what the connected controller's primary button does for the linked seat.

Possible values:

- **Request:** Triggers the seat's Request action.
- **Release:** Triggers the seat's Release action.
- **Mute:** Toggles the seat's Mute control.

WEB SERVER

The "Web Server" section appears only when TSC Type is set to `Web`. It serves a self-contained browser-based control page for the seat, requiring no physical touch screen controller. A gear icon

on the control page opens a small settings menu with a "Language" selector (a per-browser override of the plugin's configured Language property, remembered via local storage) and a Logout button.

Web Interface

Controls name: `ConfigurationWebInterface`
Type: `Text-ComboBox`
Conditional: `Mode` is `Single Seat` and `TSC Type` is `Web`.
Pin Style: `Input` / `Output`
Pin Name: `Configuration~Web Interface`
Description: Selects the network interface the embedded web server listens on, or `Any` to listen on all interfaces. This selection only affects which IP address is shown in the "URL" field below; the web server always listens on all interfaces.

Web Buttons

Controls name: `ConfigurationWebMode`
Type: `Text-ComboBox`
Conditional: `Mode` is `Single Seat` and `TSC Type` is `Web`.
Pin Style: `Input` / `Output`
Pin Name: `Configuration~Web Mode`
Description: Selects which button(s) the browser control page shows.

Possible values:

- **Request:** Shows only the Request button.
- **Release:** Shows only the Release button.
- **Request and Release:** Shows both the Request and Release buttons.
- **Mute:** Shows only a Mute toggle button.

Web Show Name

Controls name: `ConfigurationWebShowName`
Type: `Button-Toggle`
Conditional: `Mode` is `Single Seat` and `TSC Type` is `Web`.
Pin Style: `Input` / `Output`
Pin Name: `Configuration~Web Show Name`
Description: Shows or hides the seat's name on the browser control page.

Web Show State

Controls name: `ConfigurationWebShowState`
Type: `Button-Toggle`
Conditional: `Mode` is `Single Seat` and `TSC Type` is `Web`.
Pin Style: `Input` / `Output`
Pin Name: `Configuration~Web Show State`

Description: Shows or hides the seat's current state (e.g., Idle, In Request List, Active, Focus) on the browser control page.

Web URL

Controls name: `ConfigurationWebURL`
Type: `Indicator-Text`
Conditional: `Mode` is `Single Seat` and `TSC Type` is `Web`.
Pin Style: `Output`
Pin Name: `Configuration~Web URL`
Description: Displays the URL to browse to in order to open this seat's control page, based on the selected Web Interface.

USER LIBRARY

The "User Library" section appears only when the User Library property is enabled. It lets this seat connect to a User Library Mode instance and recall a delegate's profile by PIN code or NFC tag.

Library Component

Controls name: `ConfigurationLibraryComponent`
Type: `Text-ComboBox`
Conditional: `User Library` property enabled.
Pin Style: `Input` / `Output`
Pin Name: `Configuration~Library Component`
Description: Selects the User Library instance to connect to for delegate recall. Only components running in `User Library` mode with Script Access set to `Script` or `All` appear in this list.

Library Recall Mode

Controls name: `ConfigurationLibraryRecallMode`
Type: `Text-ComboBox`
Conditional: `User Library` property enabled.
Pin Style: `Input` / `Output`
Pin Name: `Configuration~Library Recall Mode`
Description: Selects what a successful recall applies to the seat.

Possible values:

- **Keep Seat Configuration:** Only the seat's name is updated; existing Role, Priority, Gain, mixes, and vote weight are left unchanged.
- **Load User Configuration:** The seat's Name, Role, Priority, Gain, Floor Mix, Group Mixes A-H, and vote weight are all overwritten with the values stored in the recalled delegate's profile.

Library Recall Pin

Controls name: `ConfigurationLibraryPin`
Type: `Text-Text Edit`
Conditional: `User Library` property enabled.

Pin Style: `Input` / `Output`
Pin Name: `Configuration~Library Pin`
Description: Typing a PIN code and losing focus (or presenting a matching NFC tag on the connected sensors component) automatically recalls the matching delegate's profile, then clears this field. Also enables the seat automatically.

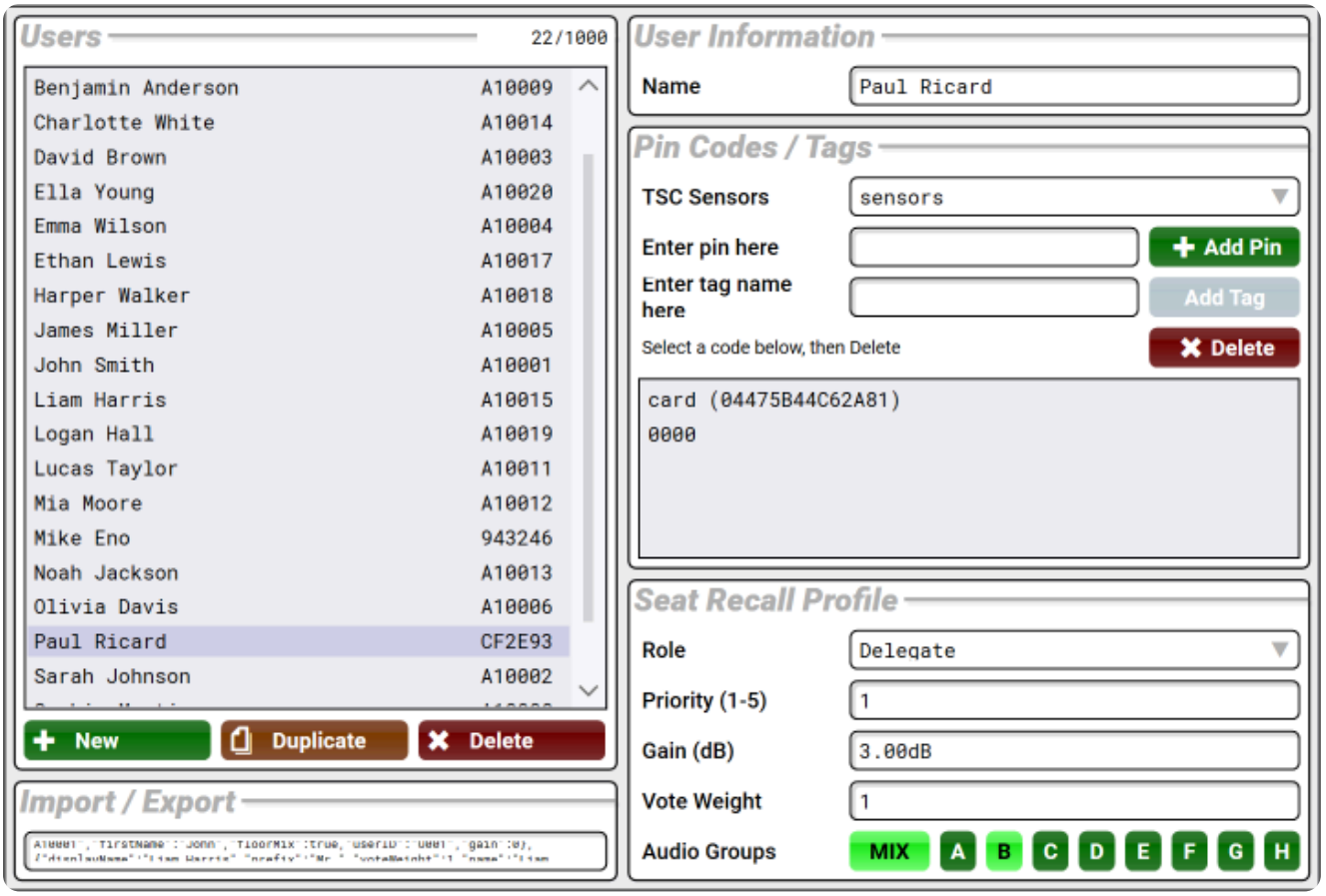
Library Recall

Controls name: `ConfigurationLibraryRecall`
Type: `Button-Trigger`
Conditional: `User Library` property enabled.
Pin Style: `Input` / `Output`
Pin Name: `Configuration~Library Recall`
Description: Manually recalls the profile for the code currently in the "Library Recall Pin" field.

Library Logout

Controls name: `ConfigurationLibraryLogout`
Type: `Button-Trigger`
Conditional: `User Library` property enabled.
Pin Style: `Input` / `Output`
Pin Name: `Configuration~Library Logout`
Description: Clears the seat's name and disables the seat, logging out the currently recalled delegate.

User Library Mode



Setting Mode to `User Library` turns the plugin instance into a standalone, JSON-backed delegate directory instead of a conferencing/voting instance. Its only page is "Library", and it exposes no audio, camera, or voting pins, this instance simply stores delegate profiles for other components to recall.

Any number of Single Seat instances can connect to the same User Library instance (see User Library under Settings – Seat mode) to look up a delegate by PIN code or NFC tag and load their profile onto the seat automatically.

LIBRARY – USERS

The left-hand "Users" list shows every delegate stored in this Library instance.

- **New:** Creates a new, blank delegate entry.
- **Duplicate:** Creates a copy of the selected delegate, including its Seat Recall Profile, but never copies its PIN codes/NFC tags (codes/tags must always be unique across the whole library).
- **Delete:** Removes the selected delegate.

LIBRARY – IMPORT / EXPORT

A read-only field showing the entire delegate directory as JSON. The full contents can be copied out (e.g., for backup) and pasted into another User Library instance's field to replace its directory.

LIBRARY – USER INFORMATION

Sets the display Name for the selected delegate.

LIBRARY – PIN CODES / TAGS

Each delegate can have any number of PIN codes and/or NFC tags, both are stored together in the same list and are enforced unique across every delegate in the library, not just the selected one.

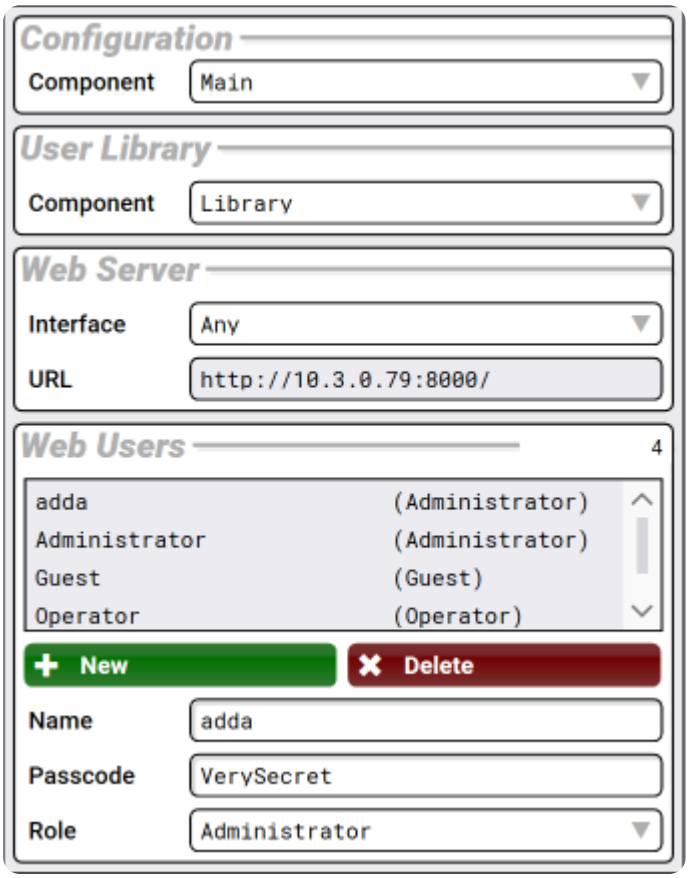
- **Name + Add:** Type an optional friendly name, then a PIN code, and press Add to register a typed code.
- **Add Tag:** Arms the button (label changes to "Waiting...") for up to 15 seconds, then registers the next tag presented to the connected TSC Sensors component (selected via the "Sensors Component" selector) using the friendly name field.
- **Delete:** Removes the selected code/tag from the list.

Attempting to register a PIN/tag that is already assigned to any delegate flashes "Duplicate!" on the Add button and the code is rejected.

LIBRARY – SEAT RECALL PROFILE

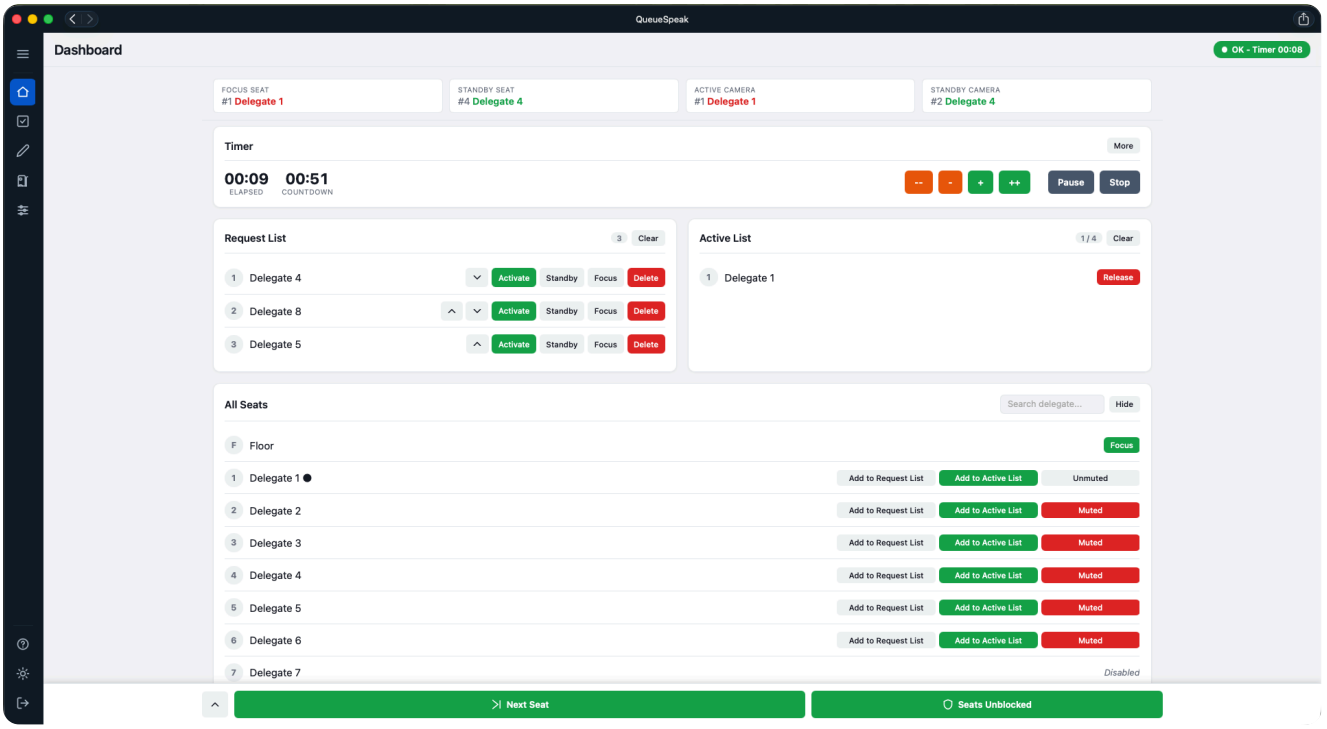
Defines what gets applied to a Single Seat instance when this delegate is recalled with "Load User Configuration" selected as the Recall Mode: Role, Priority, Gain, Floor Mix, Group Mixes A-H, and Vote Weight (mirrors the equivalent per-seat controls described under Seats [1] - N).

Dashboard Web UCI Mode



Setting Mode to `Dashboard Web UCI` turns the plugin instance into a standalone browser-based moderator/administrator dashboard. Like Single Seat mode, it connects to a Normal-mode instance via the "Component" selector and requires that instance's Script Access property to be set to `Script` or `All`; the Dashboard Web UCI instance itself does not need Script Access enabled. Its only Q-SYS-side page is "Configuration" (see Settings – Dashboard Web UCI mode below); the moderator/administrator experience itself is delivered entirely through the web browser.

WEB APPLICATION



Browsing to the instance's URL opens a single-page web app with a hamburger side-drawer for navigation:

- **Dashboard:** Shows Focus/Standby seat status (with camera info when Camera Control is enabled), the Request List and Active List (selectable, with the same Set Focus/Set Active/Set Standby/Delete actions available on the physical Dashboard), an optional searchable per-seat delegate grid (Add to Request List / Add to Active List, plus a per-seat Mute toggle, when Dashboard Seats Controls is enabled), and the Timer card (Start/Pause/Stop, plus an expandable settings panel to edit the countdown duration and the four step-size shortcuts live). The list display mode and position toggles (Dashboard list display, Position in request list, Position on seat buttons) remain physical Q-SYS controls only and are not exposed in the web app.
- **Vote (Vote Dashboard):** Mirrors the physical Vote Dashboard page, adding a turnout/leading summary and a live results view switchable between bar and pie charts, plus a one-click PDF report download once a poll is completed; shown only when Voting is enabled.
- **Vote Editor:** Mirrors the physical Vote Editor page (poll list management, options, weighting, anonymity, timer, release/request actions), shown only when Voting is enabled and only accessible to administrators.

- **User Library:** Mirrors the Library page of a connected User Library Mode instance (Users list, Import/Export, User Information, Pin Codes/Tags, Seat Recall Profile), shown only when a User Library instance is selected under Settings – Dashboard Web UCI mode and only accessible to administrators.
- **Settings:** Operators see only the Active Preset selector. Administrators additionally see the full preset/role-profile editor, mirroring the physical Settings – Default mode page.

The web app updates live over a WebSocket connection (no manual refresh needed) and reconnects automatically if the connection drops, showing a "Connection lost - reconnecting..." banner for a dropped connection or a "You're offline" banner if the browser itself loses network access; an "Update available" banner with a one-click Reload button appears when a newer version of the interface is ready. The drawer footer groups a "Sessions connected" badge (shown only when more than one browser session is connected at once), a "Language" selector, and a "Theme" toggle, followed below a divider by a single "Help" entry that opens the Help popup, and finally a "Session" control to log out and return to the lock screen.

HELP POPUP

Opened from the drawer's "Help" entry, the icon-rail's command button, or by pressing `?` at any time, the Help popup regroups the Keyboard Shortcuts list, the "Take a tour" button, and a small About section (plugin name, version, and manufacturer) in one place.

Keyboard Shortcuts: Page-switching (`F1` - `F5`) works from anywhere; the rest apply only while viewing the Dashboard page, are ignored while typing in a field, and are disabled while the lock screen, a modal, or the guided tour is open.

Key	Action
`Space`	Next Seat
`B`	Block / Unblock seats
`R`	Reset All
`T`	Play / Pause timer
`S`	Stop timer
`↑` / `↓`	Timer + / Timer -
`Shift` + `↑` / `↓`	Timer ++ / Timer --
`F1` - `F5`	Switch pages
`?`	Show this help

Guided Tour: The "Take a tour" button closes the Help popup and spotlights each major part of the interface in turn with a short explanation, dimming the rest of the screen around the highlighted

control. It automatically walks through every page the signed-in session can see - Dashboard, Vote Dashboard, Vote Editor, User Library, and Settings - skipping anything hidden or not applicable, such as the Vote pages when Voting is disabled, the Vote Editor/User Library/preset editor/per-preset configuration groups when signed in as Operator, or the Video/Auto Camera Switch settings groups when Camera Control is disabled.

Use the on-screen Next/Previous/Skip buttons, or the keyboard: `Space`, `→`, or `Enter` to advance, `←` to go back, and `Escape` to close the tour at any time.

About: A short read-only block showing the plugin name, running version, and manufacturer.

KIOSK MODE

Kiosk Mode turns the Dashboard page into a chrome-free, read-only display suited to a public screen or an unattended monitor: the page title, status indicator, side navigation, quick actions, and all interactive buttons/inputs are hidden, the seat-status area shows only the Countdown timer (enlarged, with the Elapsed time and the per-seat "All Seats" grid hidden), and the page automatically switches to the Vote page while a poll is in progress, returning to the Dashboard once it ends. A small floating exit button in the corner returns to the normal interface.

Kiosk Mode can be activated by opening the instance's URL with a `?kiosk=1` query parameter (`?kiosk=0` to force it off); the setting is remembered per browser via local storage. It is not exposed as a toggle in the drawer, but a Guest session (see Access levels) is always forced into it regardless of this setting.

LANGUAGE SELECTOR

The drawer's "Language" selector lets a browser session display the interface in a different language than the plugin's configured Language property, without affecting anyone else connected to the same instance or the language used for generated vote reports. The choice is remembered per browser via local storage.

ACCESS LEVELS

The web app is protected by a lock screen requiring a passcode before anything is shown. The passcode is matched against the Web Users list to determine the session's role and display name. Three roles are supported:

- **Guest:** Kiosk-mode only - a fully read-only display (no navigation, no interactive controls); the server also rejects any action for this role, so it cannot be bypassed from the browser. The floating corner button (normally used to exit Kiosk Mode) logs the guest out instead, since kiosk mode cannot be turned off for this role.
- **Operator:** Can use the Dashboard and Vote Dashboard pages, and change the Active Preset from Settings.
- **Administrator:** Can additionally use the Vote Editor, User Library, and Web Users pages, and the full preset/role-profile editor on the Settings page.

The signed-in user's name is shown in the drawer next to the Logout button. If the Web Users list is empty, the web app is fully open with administrator access and no lock screen prompt is required.

Settings – Dashboard Web UCI mode

COMPONENT

Controls name: `ConfigurationComponent`
Type: `Text-ComboBox`
Pin Style: `Input` / `Output`
Pin Name: `Configuration~Component`
Description: Displays available QueueSpeak plugins in "Normal" mode; select the plugin to connect to. To appear in this list, the Normal-mode instance must have "Script access" set to "ALL" or "Script".

USER LIBRARY

Controls name: `ConfigurationLibraryComponent`
Type: `Text-ComboBox`
Pin Style: `Input` / `Output`
Pin Name: `Configuration~Library Component`
Description: Selects the User Library Mode instance to connect to, exposing its directory as the "User Library" page in the web app. Only components running in `User Library` mode with Script Access set to `Script` or `All` appear in this list. Optional - leave unselected if delegate directory management is not needed from this dashboard.

WEB INTERFACE

Controls name: `ConfigurationWebInterface`
Type: `Text-ComboBox`
Pin Style: `Input` / `Output`
Pin Name: `Configuration~Web Interface`
Description: Selects the network interface the embedded web server listens on, or `Any` to listen on all interfaces. This selection only affects which IP address is shown in the "Web URL" field below; the web server always listens on all interfaces.

WEB URL

Controls name: `ConfigurationWebURL`
Type: `Indicator-Text`
Pin Style: `Output`
Pin Name: `Configuration~Web URL`
Description: Displays the URL to browse to in order to open the Dashboard Web UCI, based on the selected Web Interface.

WEB USERS

Manages the list of accounts allowed to log into the Dashboard Web UCI web app. Users can be added, edited, and deleted either from this Controls page section or from the Administrator's Settings page in the web app itself - both stay in sync. Defaults to one Administrator and one Operator user for a fresh install. See Access levels for what each role can do.

Controls name: `ConfigurationWebUsers`
Type: `Text-Text Edit`
Pin Style: `None`
Description: Stores the full user list as JSON. Not meant to be hand-edited - use the fields below or the web app's Settings page instead.

Controls name: `ConfigurationWebUserCount`
Type: `Indicator-Text`
Pin Style: `None`
Description: Displays the number of configured web users.

Controls name: `ConfigurationWebUserList`
Type: `Text-List Box`
Pin Style: `None`
Description: Selects a user to edit below. Each row shows the user's name and role.

Controls name: `ConfigurationWebUserAdd`
Type: `Button-Trigger`
Pin Style: `None`
Description: Creates a new user (default name "New User", `Operator` role, empty passcode) and selects it for editing.

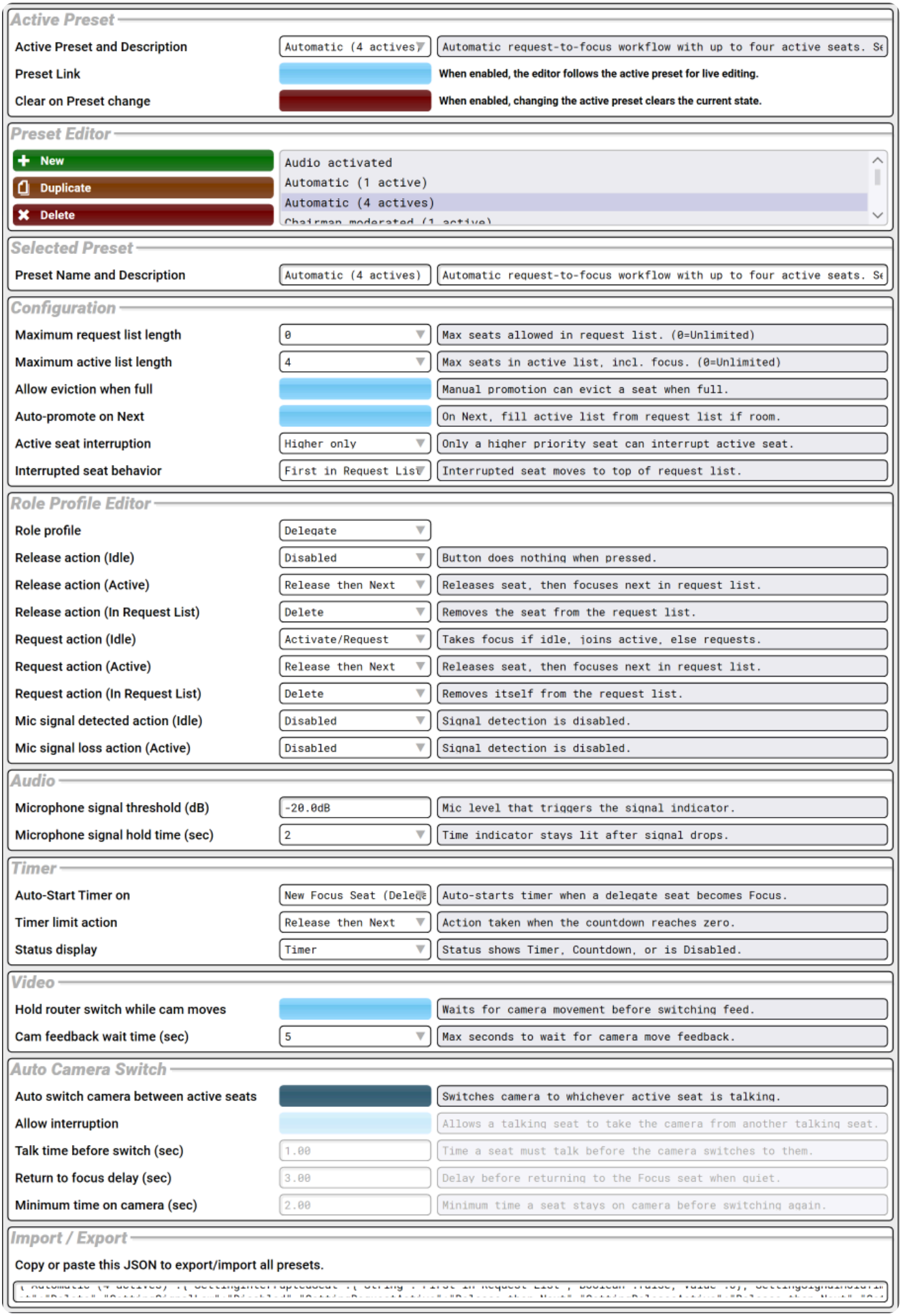
Controls name: `ConfigurationWebUserDelete`
Type: `Button-Trigger`
Pin Style: `None`
Description: Deletes the selected user. Disabled when no user is selected.

Controls name: `ConfigurationWebUserName`
Type: `Text-Text Edit`
Pin Style: `None`
Description: Sets the display name shown for the selected user, including next to the Logout button in the web app's drawer once signed in.

Controls name: `ConfigurationWebUserPasscode`
Type: `Text-Text Edit`
Pin Style: `None`
Description: Sets the passcode the selected user enters on the web app's lock screen.

Controls name: `ConfigurationWebUserRole`
Type: `Text-ComboBox`
Pin Style: `None`
Description: Sets the selected user's access level: `Guest`, `Operator`, or `Administrator`.

Settings – Default mode


Active Preset
Active Preset and Description
Automatic (4 actives)
Automatic request-to-focus workflow with up to four active seats. Se
Preset Link
When enabled, the editor follows the active preset for live editing.
Clear on Preset change
When enabled, changing the active preset clears the current state.
Preset Editor
New
Duplicate
Delete
Audio activated
Automatic (1 active)
Automatic (4 actives)
Selected Preset
Preset Name and Description
Automatic (4 actives)
Automatic request-to-focus workflow with up to four active seats. Se
Configuration
Maximum request list length
0
Max seats allowed in request list. (0=Unlimited)
Maximum active list length
4
Max seats in active list, incl. focus. (0=Unlimited)
Allow eviction when full
Manual promotion can evict a seat when full.
Auto-promote on Next
On Next, fill active list from request list if room.
Active seat interruption
Higher only
Only a higher priority seat can interrupt active seat.
Interrupted seat behavior
First in Request List
Interrupted seat moves to top of request list.
Role Profile Editor
Role profile
Delegate
Release action (Idle)
Disabled
Button does nothing when pressed.
Release action (Active)
Release then Next
Releases seat, then focuses next in request list.
Release action (In Request List)
Delete
Removes the seat from the request list.
Request action (Idle)
Activate/Request
Takes focus if idle, joins active, else requests.
Request action (Active)
Release then Next
Releases seat, then focuses next in request list.
Request action (In Request List)
Delete
Removes itself from the request list.
Mic signal detected action (Idle)
Disabled
Signal detection is disabled.
Mic signal loss action (Active)
Disabled
Signal detection is disabled.
Audio
Microphone signal threshold (dB)
-20.0dB
Mic level that triggers the signal indicator.
Microphone signal hold time (sec)
2
Time indicator stays lit after signal drops.
Timer
Auto-Start Timer on
New Focus Seat (Delega
Auto-starts timer when a delegate seat becomes Focus.
Timer limit action
Release then Next
Action taken when the countdown reaches zero.
Status display
Timer
Status shows Timer, Countdown, or is Disabled.
Video
Hold router switch while cam moves
Waits for camera movement before switching feed.
Cam feedback wait time (sec)
5
Max seconds to wait for camera move feedback.
Auto Camera Switch
Auto switch camera between active seats
Switches camera to whichever active seat is talking.
Allow interruption
Allows a talking seat to take the camera from another talking seat.
Talk time before switch (sec)
1.00
Time a seat must talk before the camera switches to them.
Return to focus delay (sec)
3.00
Delay before returning to the Focus seat when quiet.
Minimum time on camera (sec)
2.00
Minimum time a seat stays on camera before switching again.
Import / Export
Copy or paste this JSON to export/import all presets.

ACTIVE PRESET SELECTOR

Controls name: `PresetSelector`
Type: `Text-ComboBox`
Pin Style: `Input` / `Output`
Pin Name: `Settings~Active Preset~Selector`
Description: Selects the preset to use as configuration template for the plugin.

The plugin ships with editable default templates. They are provided only as starting points: users can modify, duplicate, or delete them, and the plugin is designed so every workflow can be fully configured for the room, moderator, and participants.

Default templates:

- **Automatic (4 actives):** Automatic request-to-focus workflow with up to four active seats. Seat buttons can request, release, and advance to the next request automatically.
- **Automatic (1 active):** Automatic single-speaker workflow. One active seat is allowed, and release actions advance to the next requested seat.
- **Chairman moderated (4 actives):** Chairman-moderated workflow with up to four active seats. Seat requests are queued, and the chairman or operator chooses which seat becomes active or focused.
- **Chairman moderated (1 active):** Chairman-moderated single-speaker workflow. Seat requests are queued for chairman or operator selection, with only one active seat allowed.
- **Audio activated:** Voice-activated workflow. Microphone signal can activate or queue a seat, and signal loss releases the active seat and advances to the next request.
- **Open discussion:** Multi-speaker discussion workflow with no active-list limit. Seats can join the Active List directly and release themselves when finished.
- **Off:** All seat requests are disabled. No active seats are allowed, and all request actions are ignored.

ACTIVE PRESET DESCRIPTION

Controls name: `PresetSelectorDescription`
Type: `Indicator-Text`
Pin Style: `Output`
Pin Name: `Settings~Active Preset~Description`
Description: Shows the description of the currently selected active preset.

PRESET LINK

Controls name: `PresetLink`
Type: `Button-Toggle`
Pin Style: `Input` / `Output`
Pin Name: `Settings~Preset~Link`
Description: When enabled, the preset editor follows the active preset for live editing.

CLEAR ON PRESET CHANGE

Controls name: `PresetClearOnChange`
Type: `Button-Toggle`
Pin Style: `Input` / `Output`
Pin Name: `Settings~Preset~Clear on Preset change`
Description: When enabled, changing the active preset clears the current Focus, Active List, Request List, timer, and runtime state after applying the selected preset settings.

PRESET LIST

Controls name: `PresetEditor`
Type: `Text-ListBox`
Pin Style: `None`
Description: Displays the presets available in the preset editor.

PRESET NEW

Controls name: `PresetAdd`
Type: `Button-Trigger`
Pin Style: `None`
Description: Creates a new preset.

PRESET DUPLICATE

Controls name: `PresetDuplicate`
Type: `Button-Trigger`
Pin Style: `None`
Description: Duplicates the selected preset.

PRESET DELETE

Controls name: `PresetDelete`
Type: `Button-Trigger`
Pin Style: `None`
Description: Deletes the selected preset.

PRESET NAME

Controls name: `PresetName`
Type: `Text-Text Edit`
Pin Style: `None`
Description: Edits the selected preset name.

PRESET DESCRIPTION

Controls name: `PresetDescription`
Type: `Text-Text Edit`
Pin Style: `None`
Description: Edits the selected preset description.

MAXIMUM REQUEST LIST LENGTH

Controls name: `SettingMaxRequestList`
Type: `Knob-Integer`
Pin Style: `Input` / `Output`
Pin Name: `Settings~Configuration~Maximum request list length`
Description: Sets the maximum number of seats allowed in the Request List. Use 0 for unlimited. When the maximum is reached, new requests are ignored until a Request List slot becomes available.

MAXIMUM ACTIVE LIST LENGTH

Controls name: `SettingMaxActiveList`
Type: `Knob-Integer`
Pin Style: `Input` / `Output`
Pin Name: `Settings~Configuration~Maximum active list length`
Description: Sets the maximum number of seats allowed in the Active List, including the Focus seat. Use 0 for unlimited.

> **Pro Tip:** Use Maximum Request List length and Maximum Active List length together to enforce FIFO-style workflows while limiting how many seats can remain active at the same time.

ALLOW EVICTION WHEN FULL

Controls name: `SettingAllowEviction`
Type: `Button-Toggle`
Pin Style: `Input` / `Output`
Pin Name: `Settings~Configuration~Allow eviction when full`
Description: When enabled, manual Set Focus and Set Active promotions can evict a seat from the Active List when Maximum Active List Length is reached. Set Focus removes the last seat in the Active List; Set Active can remove the current Focus seat. When disabled, manual promotion is blocked until an Active List slot is available.

AUTO PROMOTE ON NEXT

Controls name: `SettingAutoPromote`
Type: `Button-Toggle`
Pin Style: `Input` / `Output`
Pin Name: `Settings~Configuration~Auto-promote on Next`
Description: When enabled, the Next action can automatically fill available Active List slots from the Request List. With Maximum Active List Length above 1, each Next press tops up one seat at a time from the Request List without releasing the current Focus, until the Active List is full; once full, Next resumes releasing the current Focus and rotating in the next seat.

ACTIVE SEAT INTERRUPTION (PRIORITY)

Controls name: `SettingPriority`
Type: `Text-ComboBox`
Pin Style: `Input` / `Output`
Pin Name: `Settings~Configuration~Active seat interruption`
Description: Defines whether the active seat can be interrupted by another seat and how the plugin handles seat priority conflicts.

Possible values:

- **Disabled:** Seats cannot be interrupted by another seat, regardless of their assigned priority level.
- **Higher only:** A seat with a higher priority level than the active seat can interrupt and become the new active seat.
- **Higher/equal:** A seat with an equal or higher priority level than the active seat can interrupt and become the new active seat.

INTERRUPTED SEAT BEHAVIOUR

Controls name: `SettingInterruptedSeat`
Type: `Text-ComboBox`
Pin Style: `Input` / `Output`
Pin Name: `Settings~Configuration~Interrupted seat behavior`
Description: Defines the behavior applied to the previously active seat when it is interrupted by a higher-priority seat.

Possible values:

- **Clear:** The interrupted seat returns to idle state.
- **First in Request List:** The interrupted seat is placed at the top of the Request List, making it the next seat to receive focus.

ROLE PROFILE

Controls name: `PresetEditorRole`
Type: `Text-ComboBox`
Pin Style: `Input` / `Output`
Pin Name: `Settings~RoleProfileEditor~Role profile`
Description: Selects which seat role profile is being edited in the Role Profile Editor.

RELEASE ACTION (IDLE)

Controls name: `SettingReleaseIdle`
Type: `Text-ComboBox`
Pin Style: `Input` / `Output`
Pin Name: `Settings~RoleProfileEditor~Release action (Idle)`
Description: Defines the action triggered when the "Release" button is pressed while in the "Idle" state.

Possible values:

- **Disabled:** Disables the button.
- **Request:** Adds the seat to the Request List.
- **Activate/Request:** Takes focus if the room is idle, joins the Active List if there is room, otherwise adds the seat to the Request List.
- **Unmute:** Unmutes the seat's microphone.
- **Mute:** Mutes the seat's microphone.
- **Toggle Mute:** Toggles the microphone's mute state.
- **Override:** Gives the seat exclusive focus and mutes other seats until Override is released.

RELEASE ACTION (ACTIVE)

Controls name: `SettingReleaseActive`
Type: `Text-ComboBox`
Pin Style: `Input` / `Output`
Pin Name: `Settings~RoleProfileEditor~Release action (Active)`
Description: Defines the action triggered when the "Release" button is pressed while in the "Active" state.

Possible values:

- **Disabled:** Disables the button.
- **Release then Next:** Releases the seat, then focuses the next standby or Request List seat. If no standby seat is found, focus returns to the Floor.
- **Release then Floor:** Releases the seat, then returns focus to the Floor.
- **Unmute:** Unmutes the seat's microphone.
- **Mute:** Mutes the seat's microphone.
- **Toggle Mute:** Toggles the microphone's mute state.
- **Override:** Gives the seat exclusive focus and mutes other seats until Override is released.

RELEASE ACTION (IN REQUEST LIST)

Controls name: `SettingReleaseInRequestList`
Type: `Text-ComboBox`
Pin Style: `Input` / `Output`
Pin Name: `Settings~RoleProfileEditor~Release action (In Request List)`
Description: Defines the action triggered when the "Release" button is pressed while in the "In Request List" state.

Possible values:

- **Disabled:** Disables the button.
- **Delete:** Removes the seat from the Request List.
- **Unmute:** Unmutes the seat's microphone.
- **Mute:** Mutes the seat's microphone.
- **Toggle Mute:** Toggles the microphone's mute state.

- **Override:** Gives the seat exclusive focus and mutes other seats until Override is released.

REQUEST ACTION (IDLE)

Controls name: `SettingRequestIdle`
Type: `Text-ComboBox`
Pin Style: `Input` / `Output`
Pin Name: `Settings~RoleProfileEditor~Request action (Idle)`
Description: Defines the action triggered when the "Request" button is pressed while in the "Idle" state.

Possible values:

- **Disabled:** Disables the button.
- **Request:** Adds the seat to the Request List.
- **Activate/Request:** Takes focus if the room is idle, joins the Active List if there is room, otherwise adds the seat to the Request List.
- **Unmute:** Unmutes the seat's microphone.
- **Mute:** Mutes the seat's microphone.
- **Toggle Mute:** Toggles the microphone's mute state.
- **Override:** Gives the seat exclusive focus and mutes other seats until Override is released.

REQUEST ACTION (ACTIVE)

Controls name: `SettingRequestActive`
Type: `Text-ComboBox`
Pin Style: `Input` / `Output`
Pin Name: `Settings~RoleProfileEditor~Request action (Active)`
Description: Defines the action triggered when the "Request" button is pressed while in the "Active" state.

Possible values:

- **Disabled:** Disables the button.
- **Release then Next:** Releases the seat, then focuses the next standby or Request List seat. If no standby seat is found, focus returns to the Floor.
- **Release then Floor:** Releases the seat, then returns focus to the Floor.
- **Unmute:** Unmutes the seat's microphone.
- **Mute:** Mutes the seat's microphone.
- **Toggle Mute:** Toggles the microphone's mute state.

REQUEST ACTION (IN REQUEST LIST)

Controls name: `SettingRequestInRequestList`
Type: `Text-ComboBox`
Pin Style: `Input` / `Output`
Pin Name: `Settings~RoleProfileEditor~Request action (In Request List)`

Description: Defines the action triggered when the "Request" button is pressed while in the "In Request List" state.

Possible values:

- **Disabled:** Disables the button.
- **Delete:** Removes the seat from the Request List.
- **Unmute:** Unmutes the seat's microphone.
- **Mute:** Mutes the seat's microphone.
- **Toggle Mute:** Toggles the microphone's mute state.
- **Override:** Gives the seat exclusive focus and mutes other seats until Override is released.

MIC SIGNAL DETECTED ACTION (IDLE)

Controls name: `SettingSignalHigh`
Type: `Text-ComboBox`
Pin Style: `Input` / `Output`
Pin Name: `Settings~RoleProfileEditor~Mic signal detected action (Idle)`
Description: Defines the action triggered when a signal is detected at the microphone's input, based on the threshold defined in the Microphone signal threshold (dB) configuration. **Pro-tip:** You can use microphone signal detection as an automation trigger for Request List management.

Possible values:

- **Disabled:** Detection is disabled.
- **Request:** Adds the seat to the Request List.
- **Activate/Request:** Takes focus if the room is idle, joins the Active List if there is room, otherwise adds the seat to the Request List.
- **Unmute:** Unmute the seat's microphone.

MIC SIGNAL LOSS ACTION (ACTIVE)

Controls name: `SettingSignalLow`
Type: `Text-ComboBox`
Pin Style: `Input` / `Output`
Pin Name: `Settings~RoleProfileEditor~Mic signal loss action (Active)`
Description: Defines the action triggered when the microphone input signal drops below the threshold defined in the Microphone signal threshold (dB) configuration.

Possible values:

- **Disabled:** Detection is disabled.
- **Release then Next:** Releases the seat, then focuses the next standby or Request List seat. If no standby seat is found, focus returns to the Floor.
- **Release then Floor:** Releases the seat, then returns focus to the Floor.
- **Mute:** Mutes the seat's microphone.

MICROPHONE SIGNAL THRESHOLD (DB)

Controls name: `SettingSignalThreshold`
Type: `Knob-Integer`
Pin Style: `Input` / `Output`
Pin Name: `Settings~Audio~Microphone signal threshold (dB)`
Description: Set the signal level threshold that activates the Mic Signal LED. When the input audio signal exceeds this threshold, the plugin triggers the action specified in the "Mic Signal Detected Action (Idle)" setting.

MICROPHONE SIGNAL HOLD TIME (SEC)

Controls name: `SettingSignalHoldTime`
Type: `Knob-Integer`
Pin Style: `Input` / `Output`
Pin Name: `Settings~Audio~Microphone signal hold time (sec)`
Description: Specifies how long the Mic Signal LED remains lit after the audio signal falls below the threshold. Once this delay has elapsed without detecting a new signal above the threshold, the plugin triggers the action specified in the "Mic Signal Loss Action (Active)" setting.

Values range from 1 to 10 seconds.

AUTO-START TIMER ON

Controls name: `SettingTimerStart`
Type: `Text-ComboBox`
Pin Style: `Input` / `Output`
Pin Name: `Settings~Timer~Auto-Start Timer on`
Description: Determines when the timer starts automatically.

Possible values:

- **Disabled:** The timer does not start automatically.
- **New Focus Seat:** The timer starts automatically whenever a new seat receives focus, except while Override is active.
- **New Focus Seat (Delegate Only):** The timer starts automatically only when a Delegate seat receives focus, except while Override is active. Chairman seats do not start the timer in this mode.

TIMER REACHED LIMIT ACTION

Controls name: `SettingTimerAction`
Type: `Text-ComboBox`
Pin Style: `Input` / `Output`
Pin Name: `Settings~Timer~Timer limit action`
Description: Determines the action taken when the timer reaches its limit.

Possible values:

- **Disabled:** Nothing happens when the timer ends.

- **Release then Next:** Releases the Focus seat, then focuses the next standby or Request List seat. If no standby seat is found, focus returns to the Floor.
- **Release then Floor:** Releases the Focus seat, then returns focus to the Floor.

TIMER STATUS DISPLAY

Controls name: `SettingTimerStatus`
Type: `Text-ComboBox`
Pin Style: `Input` / `Output`
Pin Name: `Settings~Timer~Status display`
Description: Selects what the timer status area displays.

Possible values:

- **Disabled:** Timer status display is hidden or blank.
- **Timer:** Shows elapsed timer time.
- **Countdown:** Shows remaining countdown time.

HOLD ROUTER SWITCH WHILE CAM MOVES

Controls name: `SettingRouterWait`
Type: `Button-Toggle`
Conditional: `Camera Control` enabled.
Pin Style: `Input` / `Output`
Pin Name: `Settings~Video~Hold router switch while cam moves`
Description: When enabled, the router switch is delayed until the camera has finished moving.

TIME TO WAIT FOR CAM FEEDBACK (SEC)

Controls name: `SettingRouterWaitLimit`
Type: `Knob-Integer`
Conditional: `Camera Control` enabled.
Pin Style: `Input` / `Output`
Pin Name: `Settings~Video~Cam feedback wait time (sec)`
Description: Sets the delay before switching when no `move.completed` feedback is received from the camera.

Values range from 1 to 10 seconds.

AUTO CAMERA SWITCH ENABLE

Controls name: `SettingAutoCameraSwitch`
Type: `Button-Toggle`
Conditional: `Camera Control` enabled and `Internal Audio Component` enabled.
Pin Style: `Input` / `Output`
Pin Name: `Settings~AutoCameraSwitch~Auto switch camera between active seats`
Description: Enables automatic camera switching between active seats based on microphone activity.

AUTO CAMERA SWITCH ALLOW INTERRUPTION

Controls name: `SettingCameraSwitchAllowInterruption`
Type: `Button-Toggle`
Conditional: `Camera Control` enabled and `Internal Audio Component` enabled.
Pin Style: `Input` / `Output`
Pin Name: `Settings~AutoCameraSwitch~Allow interruption`
Description: Allows a newly talking active seat to take the camera from the active seat currently on camera.

When disabled, the camera stays on the current talking seat until that seat becomes quiet. The system then looks for another active seat that is already talking; if none is found, it returns the camera to the Focus seat.

AUTO CAMERA SWITCH TALK TIME

Controls name: `SettingCameraSwitchDelay`
Type: `Knob-Float`
Conditional: `Camera Control` enabled and `Internal Audio Component` enabled.
Pin Style: `Input` / `Output`
Pin Name: `Settings~AutoCameraSwitch~Talk time before switch (sec)`
Description: Sets how long an active seat must speak before the camera switches to that seat.

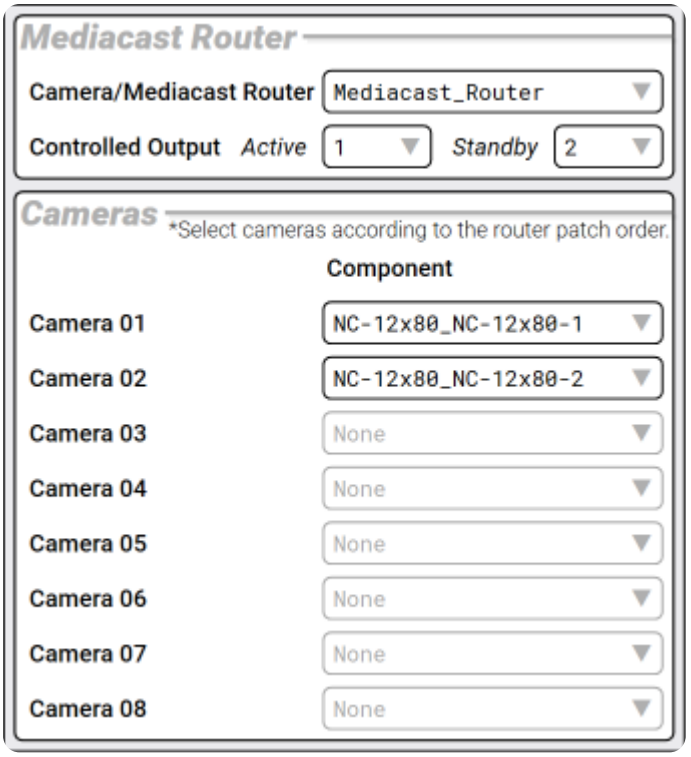
AUTO CAMERA SWITCH RETURN DELAY

Controls name: `SettingCameraSwitchReturnDelay`
Type: `Knob-Float`
Conditional: `Camera Control` enabled and `Internal Audio Component` enabled.
Pin Style: `Input` / `Output`
Pin Name: `Settings~AutoCameraSwitch~Return to focus delay (sec)`
Description: Sets how long the system waits before returning the camera to the Focus seat after the alternate active seat becomes quiet.

AUTO CAMERA SWITCH MINIMUM HOLD

Controls name: `SettingCameraSwitchMinHold`
Type: `Knob-Float`
Conditional: `Camera Control` enabled and `Internal Audio Component` enabled.
Pin Style: `Input` / `Output`
Pin Name: `Settings~AutoCameraSwitch~Minimum time on camera (sec)`
Description: Sets the minimum time a seat remains on camera before another automatic camera switch is allowed.

Cameras



Components Router Component

Controls name: `Router`
Type: `Text-ComboBox`
Conditional: `Camera Control` enabled.
Pin Style: `Input` / `Output`
Pin Name: `Cameras~Router~Component`
Description: Used to select the Mediacast router component to be used with the plugin.

COMPONENTS ROUTER ACTIVE OUTPUT

Controls name: `RouterActiveOutput`
Type: `Text-ComboBox`
Conditional: `Camera Control` enabled.
Pin Style: `Input` / `Output`
Pin Name: `Cameras~Router~Active Output`
Description: Used to select the Mediacast router output used for the active camera feed.

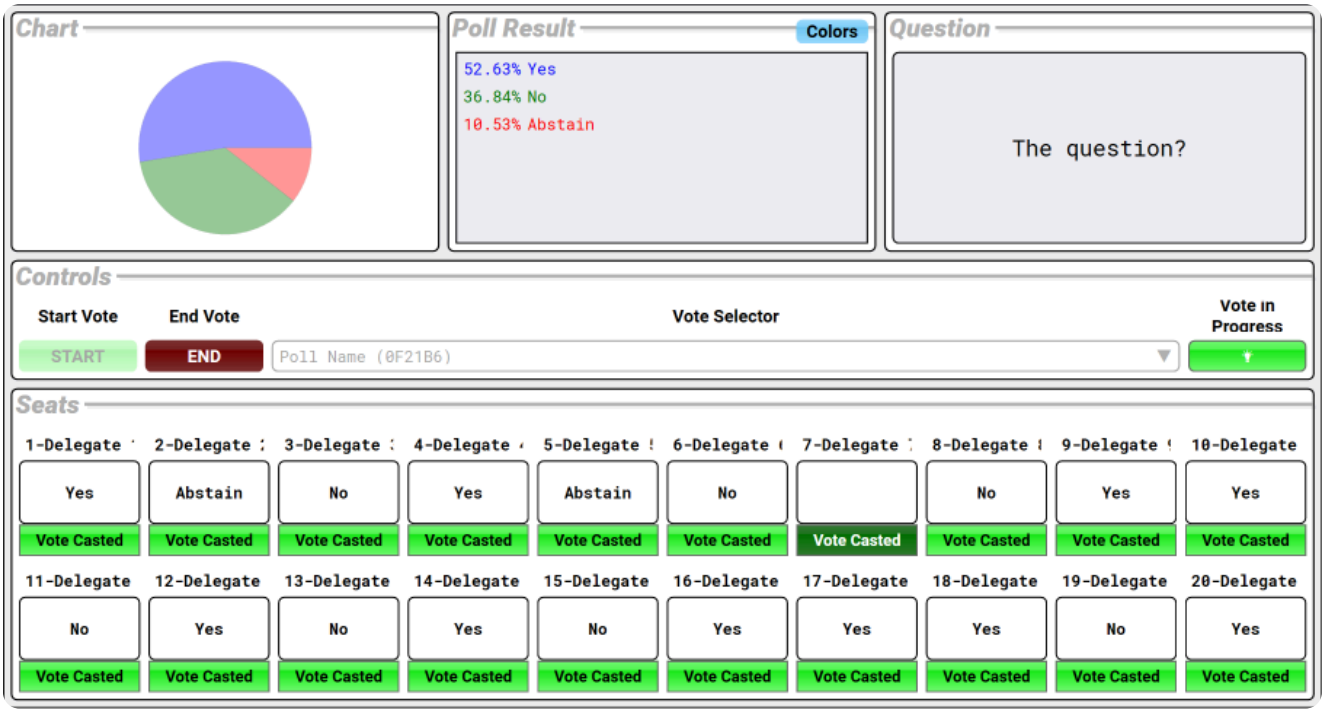
COMPONENTS ROUTER STANDBY OUTPUT

Controls name: `RouterStandbyOutput`
Type: `Text-ComboBox`
Conditional: `Camera Control` enabled.
Pin Style: `Input` / `Output`
Pin Name: `Cameras~Router~Standby Output`
Description: Used to select the Mediacast router output used for the standby camera feed.

COMPONENTS CAMERAS COMPONENT N

Controls name: `CameraN`
Type: `Text-ComboBox`
Conditional: `Camera Control` enabled.
Pin Style: `Input` / `Output`
Pin Name: `Cameras~Camera~Component N` or `Cameras~Camera~IP Address N`
Description: Used to assign cameras to the available camera slots in the plugin. In Q-SYS NC mode, select camera components. In Generic Visca mode, enter camera IP addresses. The plugin supports a maximum of 20 cameras. Select cameras following the patch order in the Mediacast router.

Vote Dashboard



Vote Dashboard SVG

Controls name: `VoteDashboardSVG`
Type: `Button-Trigger`
Conditional: `Voting` enabled.
Pin Style: `None`
Description: A real-time pie chart showing the distribution of votes cast so far, color-coded by option.

VOTE DASHBOARD RESULT

Controls name: `VoteDashboardResult`
Type: `Indicator-Text`
Conditional: `Voting` enabled.
Pin Style: `Output`

Pin Name: `Vote~Dashboard~Result`
Description: Displays the live percentage for each vote option.

VOTE DASHBOARD RESULT COLOR

Controls name: `VoteDashboardResultColor`
Type: `Button-Toggle`
Conditional: `Voting` enabled.
Pin Style: `Input` / `Output`
Pin Name: `Vote~Dashboard~Result Color`
Description: Allows toggling the color theme of the results display.

VOTE DASHBOARD TIMER

Controls name: `Timer`
Type: `Indicator-Text`
Pin Style: `Output`
Pin Name: `Timer~State~Timer`
Description: Displays the elapsed time since the current vote session started in MM:SS format.

VOTE DASHBOARD COUNTDOWN TIMER

Controls name: `Countdown`
Type: `Indicator-Text`
Pin Style: `Output`
Pin Name: `Timer~State~Countdown`
Description: Displays the remaining time before the Vote Timer Limit is reached in MM:SS format. When it reaches 00:00, the vote ends automatically only when Vote Timer Limit Action is set to `End Vote`.

VOTE DASHBOARD QUESTION

Controls name: `VoteDashboardQuestion`
Type: `Indicator-Text`
Conditional: `Voting` enabled.
Pin Style: `Output`
Pin Name: `Vote~Dashboard~Question`
Description: Shows the question text of the currently active or last completed poll.

VOTE DASHBOARD CONTROLS START [START]

Controls name: `VoteDashboardStart`
Type: `Button-StateTrigger`
Conditional: `Voting` enabled.
Pin Style: `Input` / `Output`
Pin Name: `Vote~Dashboard~Controls~Start`
Description: Initiates a new vote session using the poll selected in the Vote Selector. The timer

starts and participant voting options are activated. This button is disabled while a vote is already in progress or already completed.

VOTE DASHBOARD CONTROLS STOP [END]

Controls name: `VoteDashboardStop`
Type: `Button-StateTrigger`
Conditional: `Voting` enabled.
Pin Style: `Input` / `Output`
Pin Name: `Vote~Dashboard~Controls~Stop`
Description: Manually terminates the current vote session before the timer expires. Use this when all seats have cast their votes and there is no need to wait for the countdown. Results are finalized immediately upon ending.

VOTE DASHBOARD CONTROLS SELECTOR

Controls name: `VoteDashboardList`
Type: `Text-ComboBox`
Conditional: `Voting` enabled.
Pin Style: `Input` / `Output`
Pin Name: `Vote~Dashboard~Controls~Selector`
Description: Displays the currently selected poll configuration that will be used when starting a new vote. The selector also shows the poll's completion status. Select the desired poll from this field before pressing Start Vote.

VOTE DASHBOARD STATUS VOTE IN PROGRESS

Controls name: `VoteDashboardState`
Type: `Indicator-Led`
Conditional: `Voting` enabled.
Pin Style: `Output`
Pin Name: `Vote~Dashboard~Status~Vote in Progress`
Description: A green indicator that confirms a vote session is currently active. It turns off once the session ends or is manually stopped.

VOTE DASHBOARD SEAT N NAME

Controls name: `VoteDashboardSeatNameN`
Type: `Indicator-Text`
Conditional: `Voting` enabled and `Dashboard Seats Controls` enabled.
Pin Style: `Output`
Pin Name: `Vote~Dashboard~Seats~N~Name`
Description: Label that shows the seat's name.

VOTE DASHBOARD SEAT N VALUE

Controls name: `VoteDashboardSeatValueN`
Type: `Indicator-Text`
Conditional: `Voting` enabled and `Dashboard Seats Controls` enabled.
Pin Style: `Output`
Pin Name: `Vote~Dashboard~Seats~N~Value`
Description: Displays the option selected by this seat once a vote has been cast. Remains blank if Anonymous Vote is selected in the vote options or if the Force Anonymous Voting property is enabled.

VOTE DASHBOARD SEAT N CASTED

Controls name: `VoteDashboardSeatCastedN`
Type: `Indicator-Led`
Conditional: `Voting` enabled and `Dashboard Seats Controls` enabled.
Pin Style: `Output`
Pin Name: `Vote~Dashboard~Seats~N~Casted`
Description: A green indicator that confirms a vote has been cast for this seat.

Vote Seats

SEATS N VOTE CAST

Controls name: `VoteSeatNCast`
Type: `Button-Trigger`

Conditional: `Voting` enabled.
Pin Style: `Input` / `Output`
Pin Name: `Seats~N~Vote~Cast`
Description: Submits the vote for this seat using the current selection in the Vote Options field. If no option is selected, the button has no effect. Once a vote is cast, the button is disabled if Lock Options is enabled in the Vote Editor. If Auto Cast Option is enabled, changing the selected option casts it immediately; any still-unsubmitted eligible seats are handled automatically when the vote session ends.

SEATS N VOTE WEIGHT

Controls name: `VoteSeatNWeight`
Type: `Knob-Integer`
Conditional: `Voting` enabled.
Pin Style: `Input` / `Output`
Pin Name: `Seats~N~Vote~Weight`
Description: The numeric value applied to this seat's vote when Weighted Vote is enabled. A higher weight means this seat's vote counts for more in the final tally. Values accepted from 1 to 100.

SEATS N VOTE OPTIONS

Controls name: `VoteSeatNOptions`
Type: `Text-ComboBox`
Conditional: `Voting` enabled.
Pin Style: `Input` / `Output`
Pin Name: `Seats~N~Vote~Options`
Description: Displays the vote options available for the participant to select from. The selected value becomes the cast vote when the Cast button is pressed, when the countdown reaches zero, or when the vote ends with Lock Optionsenabled.

SEATS N VOTE VALUE

Controls name: `VoteSeatNValue`
Type: `Indicator-Text`
Conditional: `Voting` enabled.
Pin Style: `Output`
Pin Name: `Seats~N~Vote~Value`
Description: Shows the vote value recorded for this seat.

SEATS N VOTE CASTED

Controls name: `VoteSeatNCasted`
Type: `Indicator-Led`
Conditional: `Voting` enabled.
Pin Style: `Output`

Pin Name: `Seats~N~Vote~Casted`
Description: A green LED that confirms this seat's vote has been successfully registered.

Vote Editor

VOTE LIST

Controls name: `VoteList`
Type: `Text-List Box`
Conditional: `Voting` enabled.
Pin Style: `None`

Description: Displays all polls stored in the plugin with their respective name, ID, and completion status. Polls locked as templates appear at the top of the list.

VOTE ADD

Controls name: `VoteAdd`
Type: `Button-Trigger`
Conditional: `Voting` enabled.
Pin Style: `None`
Description: Creates a new blank poll configuration.

VOTE DELETE

Controls name: `VoteDelete`
Type: `Button-Trigger`
Conditional: `Voting` enabled.
Pin Style: `None`
Description: Permanently removes the selected poll. This action cannot be undone.

VOTE DUPLICATE

Controls name: `VoteDuplicate`
Type: `Button-Trigger`
Conditional: `Voting` enabled.
Pin Style: `None`
Description: Creates an editable copy of the selected poll. Useful for creating variations of an existing configuration or a template.

VOTE EXPORT

Controls name: `VoteExport`
Type: `Button-Trigger`
Conditional: `Voting` enabled.
Pin Style: `None`
Description: When pressed, exports the selected poll's results and configuration as two files, a .json and a .pdf, saved to the File section in Core Manager. The report language is controlled by the Language plugin property. If the poll is anonymous, or if Force Anonymous Voting is enabled, individual voter details are hidden from the exported files.

VOTE NAME

Controls name: `VoteName`
Type: `Text-Text Edit`
Conditional: `Voting` enabled.
Pin Style: `None`
Description: The unique name used to identify this poll configuration. It appears in the Vote Selector on the Vote Dashboard when choosing which poll to run.

VOTE QUESTION

Controls name: `VoteQuestion`
Type: `Text-Text Edit`
Conditional: `Voting` enabled.
Pin Style: `None`
Description: The question text that will be displayed to participants during the vote session. This is the main prompt participants see on their voting interface.

VOTE OPTIONS

Controls name: `VoteOptions`
Type: `Text-Text Edit`
Conditional: `Voting` enabled.
Pin Style: `None`
Description: A comma-separated list of the vote choices available to participants (e.g., Yes, No, Abstain, Maybe). These options appear on participant screens and on the Vote Dashboard results. The order entered here is the order displayed.

VOTE DEFAULT

Controls name: `VoteDefault`
Type: `Text-ComboBox`
Conditional: `Voting` enabled.
Pin Style: `None`
Description: The option that is pre-selected on participant screens when the vote begins. Participants who do not actively change their selection will have this option ready to cast. Commonly set to the most neutral option (e.g., Abstain).

VOTE READ-ONLY

Controls name: `VoteReadOnly`
Type: `Button-Toggle`
Conditional: `Voting` enabled.
Pin Style: `None`
Description: When a poll is marked as read-only, it becomes a template and cannot be directly used or modified. Templates serve as base configurations to be duplicated before use, ensuring the original settings are preserved.

VOTE LOCK OPTIONS

Controls name: `VoteLockOptions`
Type: `Button-Toggle`
Conditional: `Voting` enabled.
Pin Style: `None`
Description: When enabled, participants cannot change their vote choice after casting it. This

prevents accidental or intentional vote changes once submitted. When disabled, participants may update their vote at any time before the vote session ends.

VOTE AUTO CAST OPTION

Controls name: `VoteAutoCast`
Type: `Button-Toggle`
Conditional: `Voting` enabled.
Pin Style: `None`
Description: When enabled, the vote is automatically cast as soon as the participant selects an option, no Cast button press required. Additionally, when the vote session ends, any eligible seat that has not voted casts the Default Option on its behalf. If no Default Option is configured, the seat is removed from the voting roll. When used in conjunction with the Lock Options setting and when a default option is configured, users who want to select the Default Option (already pre-selected) can use the Cast button to immediately cast their vote, or wait for the vote to auto-cast itself when the vote ends. When disabled, votes must be submitted manually using the Cast button. Seats that have not cast a vote by the end of the session are excluded from the tally, regardless of whether a Default Option is configured.

VOTE WEIGHTED

Controls name: `VoteWeighted`
Type: `Button-Toggle`
Conditional: `Voting` enabled.
Pin Style: `None`
Description: When enabled, each seat's vote contribution is counted as the number of votes defined by its assigned Weight value (configured in the Vote Seats tab). For example, a seat with a weight of 10 adds 10 votes to the tally, while a seat with a weight of 2 adds only 2. This allows certain seats to carry more influence in the final result. When disabled, all seats count as a single vote regardless of their weight setting.

VOTE ANONYMOUS

Controls name: `VoteAnonymous`
Type: `Button-Toggle`
Conditional: `Voting` enabled.
Pin Style: `None`
Description: When enabled, the individual vote choices of each seat are hidden from the Vote Dashboard display and any report export. Only the aggregate totals and percentages are shown publicly. When disabled, each seat's vote choice is visible on the dashboard in real time. (See Poll Report Example)

VOTE CHAIRMAN CAN VOTE

Controls name: `VoteChairmanCanVote`
Type: `Button-Toggle`

Conditional: `Voting` enabled.
Pin Style: `None`
Description: Allows seats with the Chairman role to vote in this poll. Delegate seats can always vote. Technician and Observer seats cannot vote.

VOTE RELEASE ACTION

Controls name: `VoteReleaseAction`
Type: `Text-ComboBox`
Conditional: `Voting` enabled.
Pin Style: `None`
Description: Defines the action performed by the seat's release button during a vote session, allowing it to double as a voting control. Available options:

- **Disabled:** The release button performs no voting action.
- **Previous:** Each press cycles to the previous available answer.
- **Next:** Each press cycles to the next available answer.
- **Cast:** Pressing the release button casts the currently selected vote.
- **Select 1 and Cast:** A single press selects option 1 and immediately casts the vote.
- **Select 2 and Cast:** A single press selects option 2 and immediately casts the vote.

VOTE REQUEST ACTION

Controls name: `VoteRequestAction`
Type: `Text-ComboBox`
Conditional: `Voting` enabled.
Pin Style: `None`
Description: Defines the action performed by the seat's request button during a vote session, allowing it to double as a voting control. Available options:

- **Disabled:** The request button performs no voting action.
- **Previous:** Each press cycles to the previous available answer.
- **Next:** Each press cycles to the next available answer.
- **Cast:** Pressing the request button casts the currently selected vote.
- **Select 1 and Cast:** A single press selects option 1 and immediately casts the vote.
- **Select 2 and Cast:** A single press selects option 2 and immediately casts the vote.

VOTE TIMER LIMIT

Controls name: `VoteTimerLimit`
Type: `Knob-Seconds`
Conditional: `Voting` enabled.
Pin Style: `None`
Description: Sets the countdown duration for the vote session in HH:MM:SS format. The running timer and remaining countdown are both visible on the Vote Dashboard during an active vote.

Reaching zero ends the vote only when Vote Timer Limit Action is set to `End Vote`; otherwise the countdown can continue below zero until the vote is ended manually.

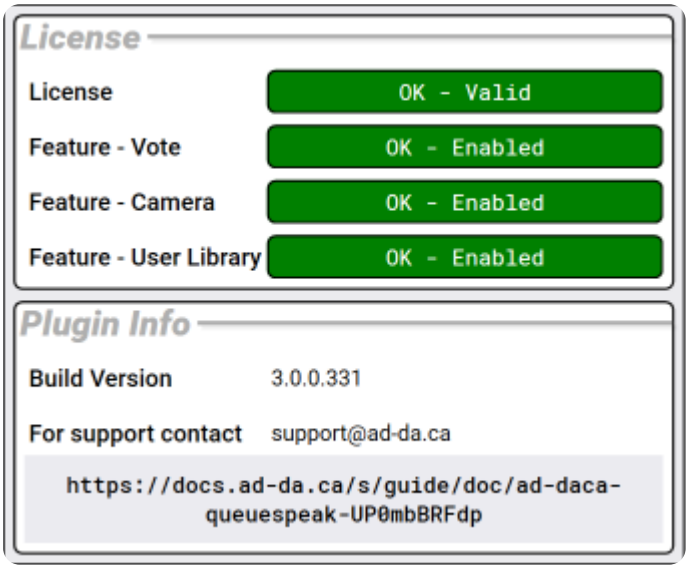
VOTE TIMER LIMIT ACTION

Controls name: `VoteTimerLimitAction`
Type: `Text-ComboBox`
Conditional: `Voting` enabled.
Pin Style: `None`
Description: Selects the action taken when the vote timer reaches zero.

Possible values:

- **Disabled:** The timer reaching zero does not automatically end the vote.
- **End Vote:** The vote session ends automatically when the timer reaches zero.

About



Read-only license status, feature status, and support information for this plugin instance.

LICENSE

Controls name: `License`
Type: `Indicator-Status`
Pin Style: `Output`
Description: Shows the validity of the license entered in the License or License Backup property - `Valid`, `Invalid`, or `Emulating` while running in Emulation mode.

FEATURE - VOTE

Controls name: `FeatureVOTE`
Type: `Indicator-Status`
Pin Style: `Output`

Description: Shows whether the licensed voting module is `Enabled` or `Disabled` for this license.

FEATURE - CAMERA

Controls name: `FeatureCAMERA`
Type: `Indicator-Status`
Pin Style: `Output`
Description: Shows whether the licensed camera module is `Enabled` or `Disabled` for this license.

FEATURE - USER LIBRARY

Controls name: `FeatureUSERS`
Type: `Indicator-Status`
Pin Style: `Output`
Description: Shows whether the licensed User Library module is `Enabled` or `Disabled` for this license.

BUILD VERSION

Displays the plugin's running build version, e.g. `3.0.0.331`. Not a Q-SYS control, this value is drawn directly from the plugin's info.

FOR SUPPORT CONTACT

Displays the support email address to contact for assistance. Not a Q-SYS control, this value is drawn directly from the plugin's info.

DOC URL

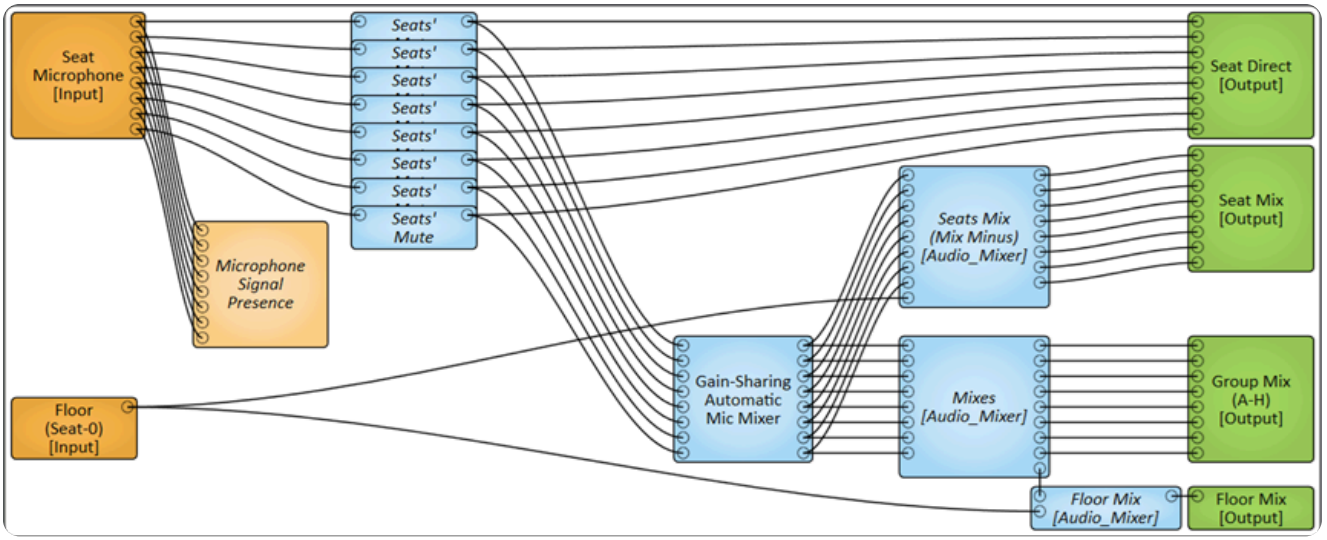
Controls name: `DocURL`
Type: `Indicator-Text`
Pin Style: `Output`
Description: Displays the URL of this user guide.

Appendix

How to rename a component

- **Q-SYS Designer 9.4 and lower:** While in offline mode, simply rename the actual component block.
- **Q-SYS Designer 9.5 and above:** While in offline mode, go to the Properties pane and change “Code Name” to the desired name and make sure “Script Access” is at “Script” or “All”.
- The “Code Name” property defines the name used by the plugin to refer to these components.

Internal Audio Schematic



> This illustrates how the internal audio schematic operates with all audio components embedded directly within the plugin.
>
> The channel count varies according to the **Seats Count** property.
>
> Seat mute functions are implemented using single-channel **Gain** blocks.
>
> An **Automatic Gain Sharing Mixer** is used to manage the floor mix.
> If you do not wish to use the internal gain-sharing mixer, you may use the **Seat Direct** output pin to retain only the internal microphone mute functionality and handle all additional processing externally.

For more information see the audio schematic section in the included Sample File.

Poll Report Example

queuespeak-v3_report.pdf


```
{
    "completed": true,
    "optionsList": [
        "Yes",
        "No",
        "Abstain"
    ],
    "name": "1-First poll_COPY",
    "readonly": true,
    "start_ts": "2026-02-19T10:24:52",
    "end_ts": "2026-02-19T10:26:09",
    "id": "7CF9A1",
```

```
    "configuration": {
        "VoteAnonymous": false,
        "VoteLockOptions": false,
        "VoteOptions": "Yes, No, Abstain",
        "VoteQuestion": "Do You?",
        "VoteReleaseAction": "Select 1 and Cast",
        "VoteTimerLimit": 120,
        "VoteAutoCast": true,
        "VoteDefault": "Abstain",
        "VoteWeighted": false,
        "VoteRequestAction": "Select 2 and Cast"
    },
    "data": {
        "voters": {
            "1": {
                "timestamp": "2026-02-19T10:26:06",
                "seatID": 1,
                "name": "Mark",
                "vote": "Yes"
            },
            "10": {
                "timestamp": "2026-02-19T10:25:56",
                "seatID": 10,
                "name": "Sofia",
                "vote": "Yes"
            },
            "4": {
                "timestamp": "2026-02-19T10:25:52",
                "seatID": 4,
                "name": "Paul",
                "vote": "Yes"
            },
            "5": {
                "timestamp": "2026-02-19T10:25:53",
                "seatID": 5,
                "name": "Alex",
                "vote": "Yes"
            },
            "2": {
                "timestamp": "2026-02-19T10:26:05",
                "seatID": 2,
                "name": "Tom",
                "vote": "Yes"
```

```
            },
            "3": {
                "timestamp": "2026-02-19T10:26:06",
                "seatID": 3,
                "name": "Travis",
                "vote": "Yes"
            },
            "8": {
                "timestamp": "2026-02-19T10:25:55",
                "seatID": 8,
                "name": "Neil",
                "vote": "No"
            },
            "9": {
                "timestamp": "2026-02-19T10:25:56",
                "seatID": 9,
                "name": "Marry",
                "vote": "No"
            },
            "6": {
                "timestamp": "2026-02-19T10:25:53",
                "seatID": 6,
                "name": "Geddy",
                "vote": "No"
            },
            "7": {
                "timestamp": "2026-02-19T10:25:52",
                "seatID": 7,
                "name": "John",
                "vote": "No"
            }
        },
        "result": {
            "simpleMajority": [
                {
                    "voteCount": 6,
                    "weightedVoteCount": 6,
                    "percentage": 60,
                    "label": "Yes"
                }
            ],
            "absoluteMajority": {
                "voteCount": 6,
```

```
                "weightedVoteCount": 6,
                "percentage": 60,
                "label": "Yes"
            }
        },
        "totalVotes": 10,
        "totalVoters": 10,
        "options": {
            "Abstain": {
                "voteCount": 0,
                "weightedVoteCount": 0,
                "percentage": 0,
                "label": "Abstain"
            },
            "Yes": {
                "voteCount": 6,
                "weightedVoteCount": 6,
                "percentage": 60,
                "label": "Yes"
            },
            "No": {
                "voteCount": 4,
                "weightedVoteCount": 4,
                "percentage": 40,
                "label": "No"
            }
        }
    }
}
```