

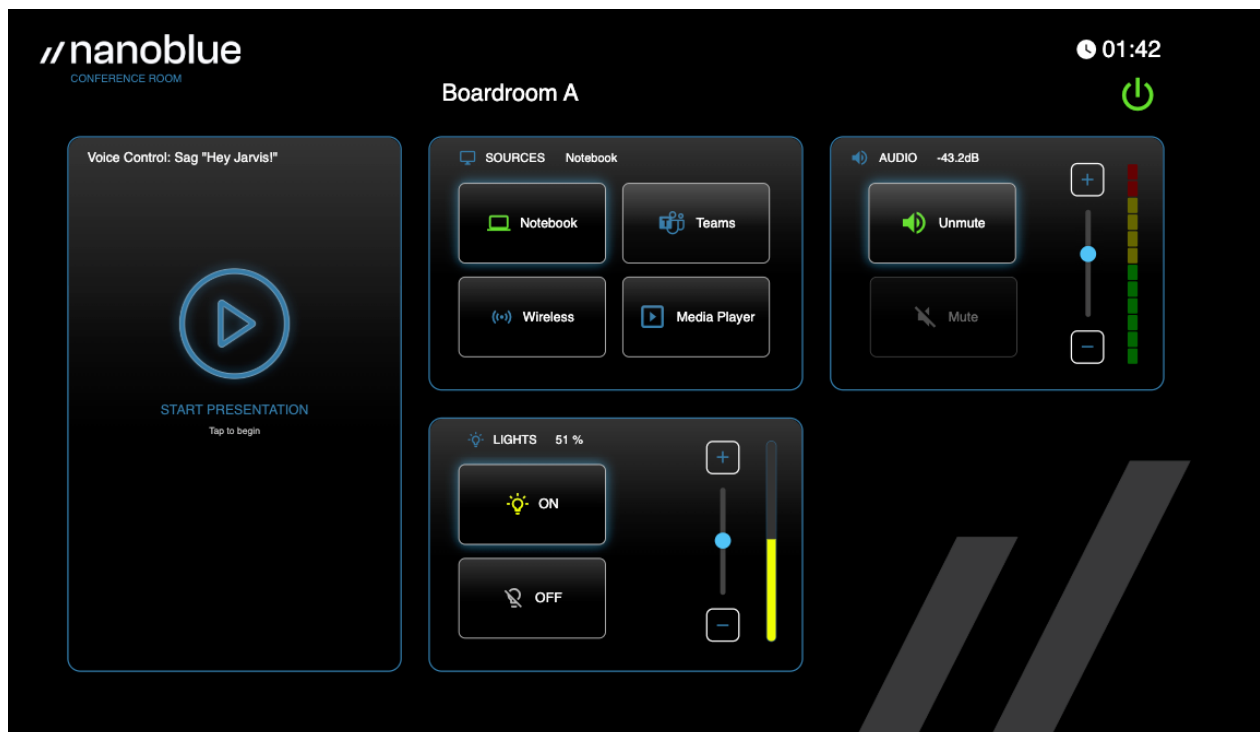


Q-SYS Control

Programming Guide for Programmers 08/2026

Contents

- 1 What it does
- 2 Q-SYS Designer prerequisites
- 3 nanoblue® Admin Dashboard
- 4 UI Studio
- 5 Widget ID Syntax
- 6 Suffix reference
- 7 End-to-end examples



1 What it does

nanoblu[®] Media Controller or Media Player acts as a **media control system** that drives a QSC Q-SYS Core directly over **QRC** (Q-SYS Remote Control). It reads and writes any control - gains, mutes, router crosspoints, snapshots, triggers - with live bidirectional feedback.

No Q-SYS plugin and no Technology Partner Program are required.

Two operator surfaces are provided:

nanoblu[®] Admin Dashboard (admin / integrator): enter the Core IP, scan the design, pick controls, operate them directly.

nanoblu[®] Interactive UI (customer): the UI Studio surface - faders, mutes, source buttons, snapshot recall - wired to Q-SYS purely by naming convention.

Everything is self-hosted in the same nanoblu[®] Editor that runs the nanoblu[®] Interactive UI. No extra broker, no cloud dependency.

2 Q-SYS Designer prerequisites

QRC can only reach what a running design exposes. Per component you want to control:

Give the component a **Code Name** (e.g. Gain1, Control_Router_8x8).

Set its **Script Access = External** – without this QRC cannot see it.

Deploy with **File → Save to Core & Run (F5)**. The first deploy matches the Core firmware to the Designer version (reboot).

The screenshot shows the Q-SYS Designer interface. On the left, the 'Design Elements' panel is visible with sections for Inventory, Schematic Pages, User Control Interfaces, and Snapshots. The main workspace displays a schematic with a 'Status' component connected to a 'Mic/Line In' component, which is then connected to two 'Gain' components (Gain 1 and Gain 2), and finally to a 'Line Out' component. A 'Gain 1 : Gain' dialog box is open, showing controls for Bypass, Invert, Mute, and Gain. On the right, the 'Properties' panel is open, showing 'Gain Properties' and 'Graphic Properties'. The 'Script Access' section is highlighted with a red circle, showing 'Code Name' as 'Gain1' and 'Script Access' as 'External'.

Gain Properties	
Max Gain (dB)	20
Min Gain (dB)	-100
Enable Ramp Controls	No

Channels	
Type	Mono

Graphic Properties	
Label	Gain 1
Position	154,122
Fill	

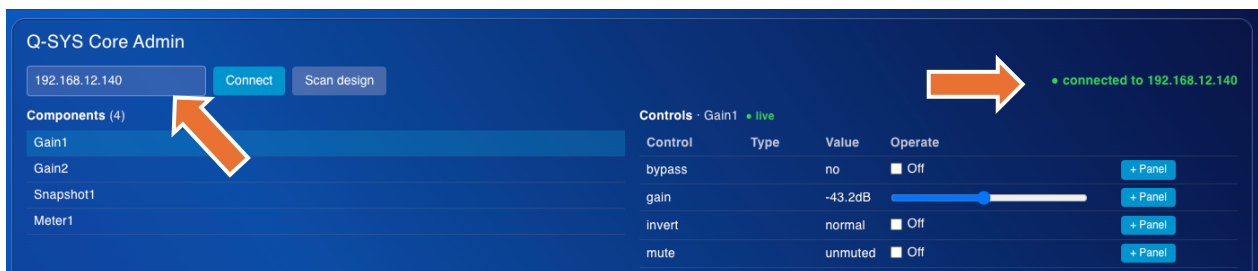
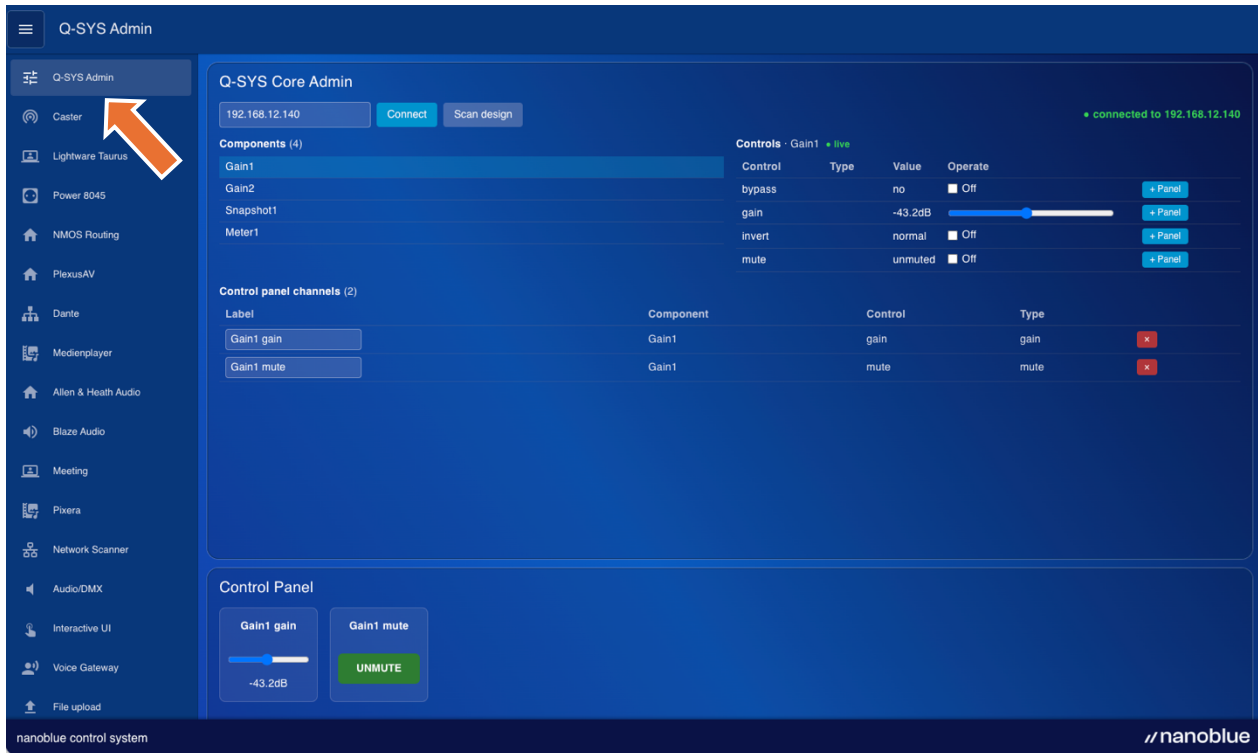
Script Access	
Code Name	Gain1
Script Access	External

Control Pins	
Bypass	☐
Gain	☐
Invert	☐
Mute	☐

3 nanoblu[®] Admin Dashboard

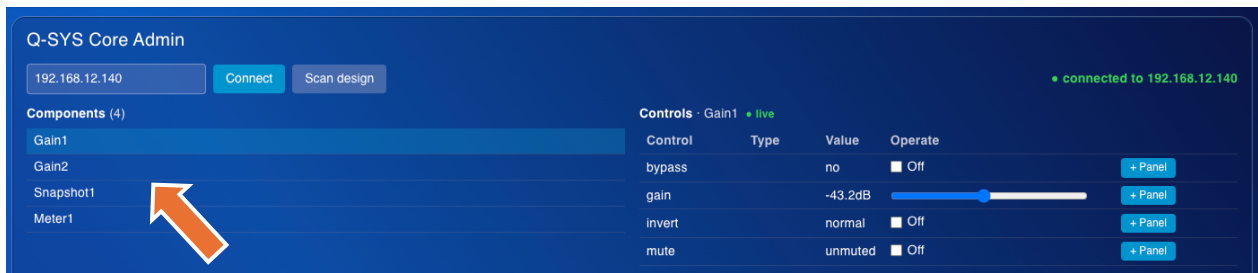
Open the nanoblu[®] Admin Dashboard using the following link:

<IP-Address>:1880/dashboard/qsyst



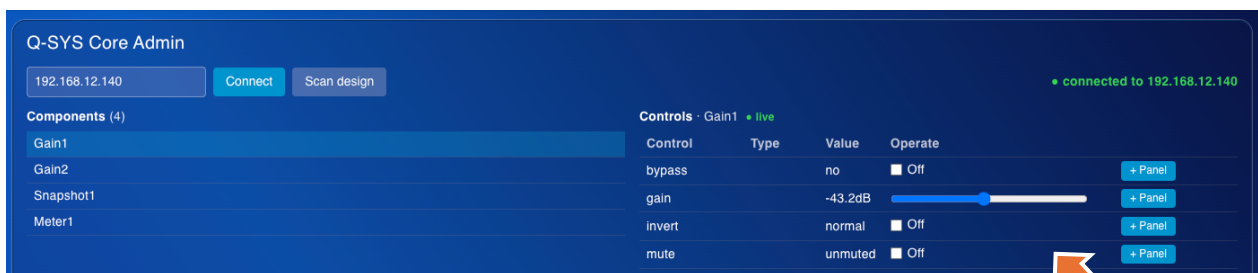
Enter the IP address of the Q-SYS Core in the top left corner and press Connect.

A successful connection will be displayed in the status message in the top right corner.



After establishing a connection, a scan for available components is automatically performed. All components for which "External" was selected in the script access are listed.

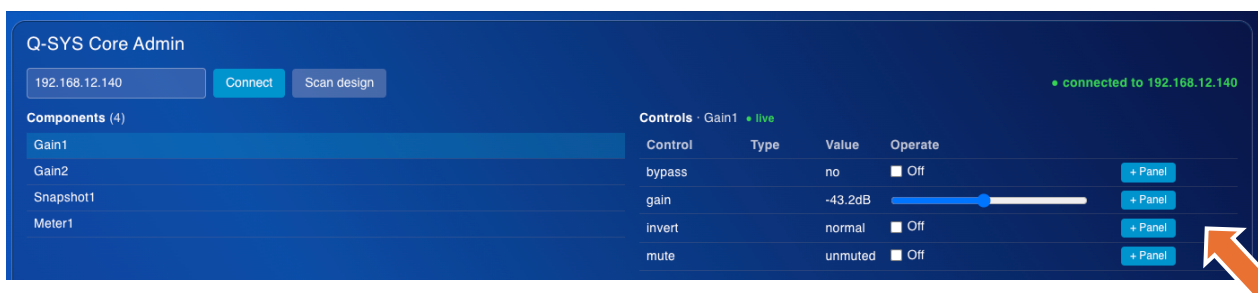
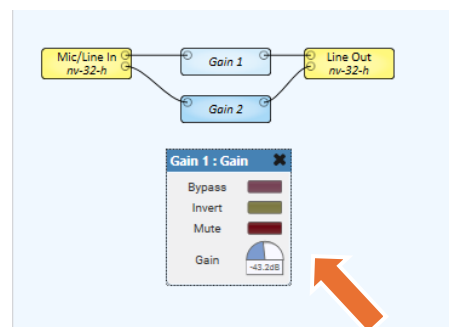
The code name entered in the field is also displayed.



Pressing a listed component displays the corresponding controls on the right.

Values can be changed by using control functions such as ON/OFF and sliders.

If you are connected to the Core in the Q-SYS Designer (F5), you can follow the live changes of the values there.



Select a control element that you want to control in the nanoblu® Interactive UI.

Pressing the "+ Panel" button adds the selected function to the List "Control panel channels" and "Control Panel" field.

Q-SYS Core Admin

192.168.12.140 Connect Scan design connected to 192.168.12.140

Components (2)

Gain1
Gain2

Controls - Gain1 live

Control	Type	Value	Operate	
bypass		no	☐ Off	+ Panel
gain		-43.2dB	<input type="range"/>	+ Panel
invert		normal	☐ Off	+ Panel
mute		unmuted	☐ Off	+ Panel

Control panel channels (2)

Label	Component	Control	Type	
Gain1 gain	Gain1	gain	gain	x
Gain1 mute	Gain1	mute	mute	x

In this list are all the functions you can use in the Interactive UI. Unnecessary functions can also be deleted here.

Control Panel

Gain1 gain

-43.2dB

Gain1 mute

UNMUTE

Again, you can follow the function live in the Q-SYS design by pressing a key or dragging a slider.

With that, all preparations are complete. Now start with integration into the Interactive UI.

Q-SYS Core Admin

192.168.12.140 Connect Scan design connected to 192.168.12.140

Components (2)

Gain1
Gain2

Controls - Gain1 live

Control	Type	Value	Operate	
bypass		no	☐ Off	+ Panel
gain		-43.2dB	<input type="range"/>	+ Panel
invert		normal	☐ Off	+ Panel
mute		unmuted	☐ Off	+ Panel

Control panel channels (2)

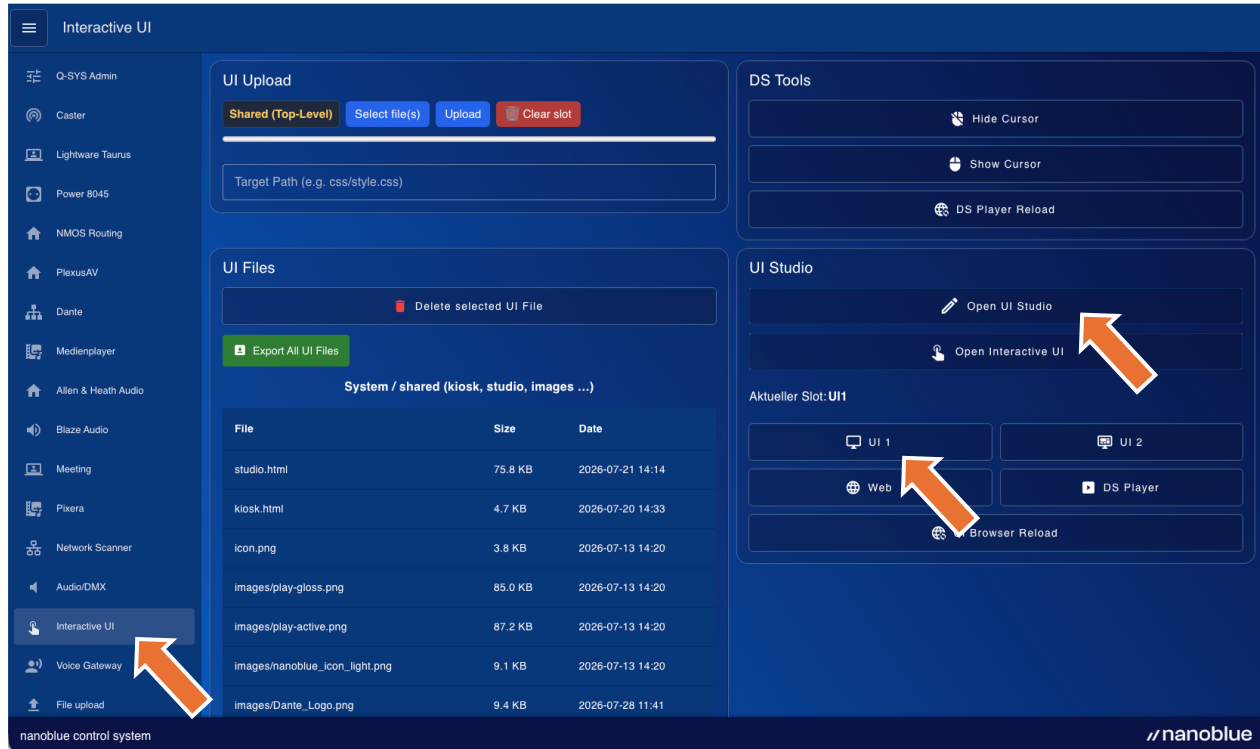
Label	Component	Control	Type	
Gain1 gain	Gain1	gain	gain	x
Gain1 mute	Gain1	mute	mute	x

Please note the labels you see in the Component and Control columns; you will need these when assigning design elements in the Interactive UI.

4 UI Studio

Now switch to the Interactive UI tab in the Admin Dashboard.

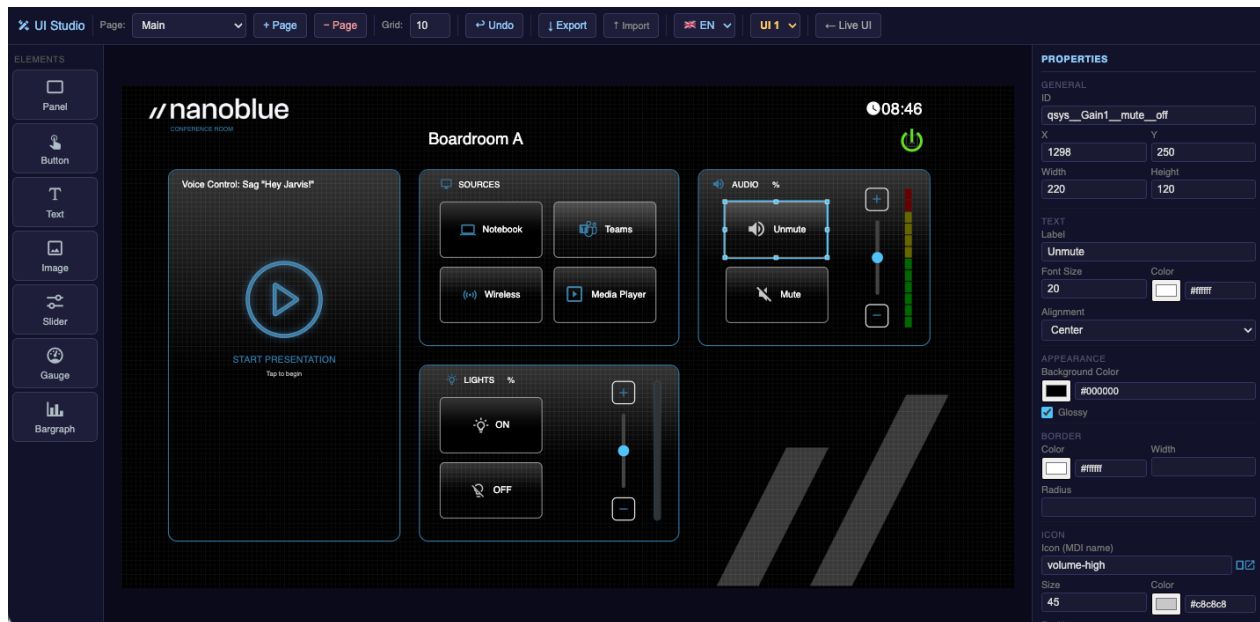
<IP-Address>:1880/dashboard/interactive



Please check if the UI slot is set to: UI 1. Now open the UI Studio.

The UI Studio can also be opened directly using this link:

<IP-Address>:1880/ui/studio.html

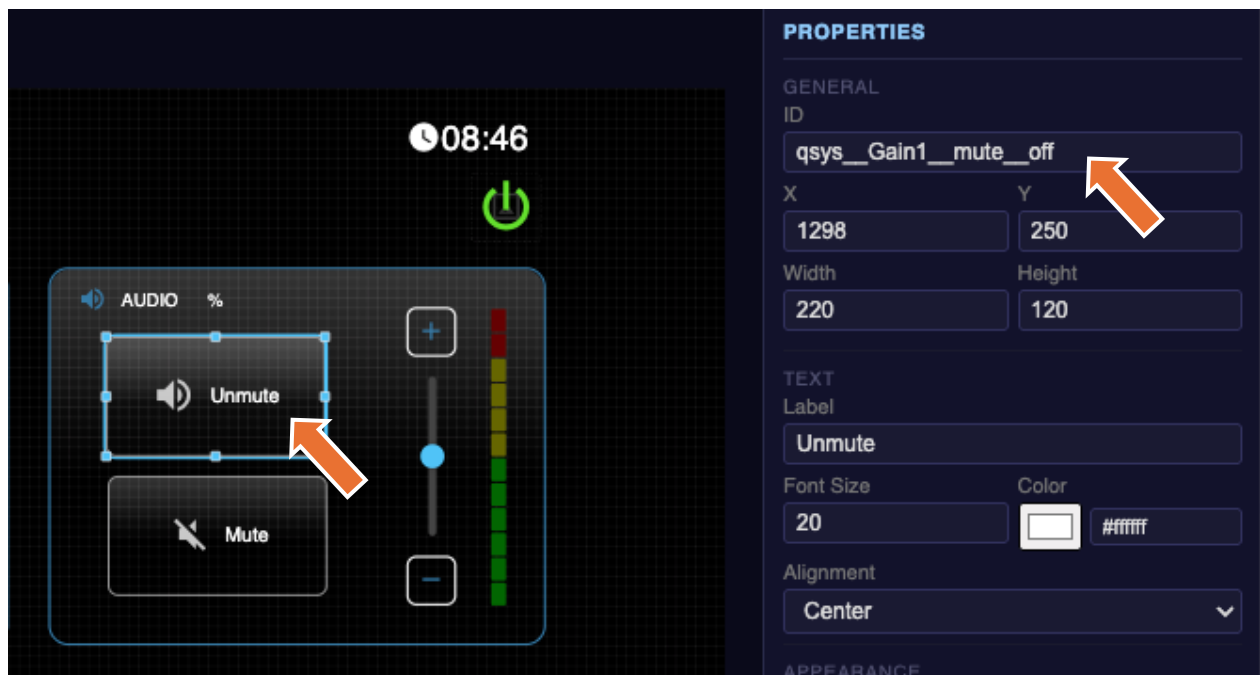


Here's a view of our demo user interface.

The entire UI can be easily created using drag and drop.

Changes are applied immediately; there's no "Save" button. The interactive UI can also be opened via this links, allowing you to follow the changes live:

<IP-Address>:1880/ui/ui1/ or <IP-Address>:1880/ui/kiosk.html



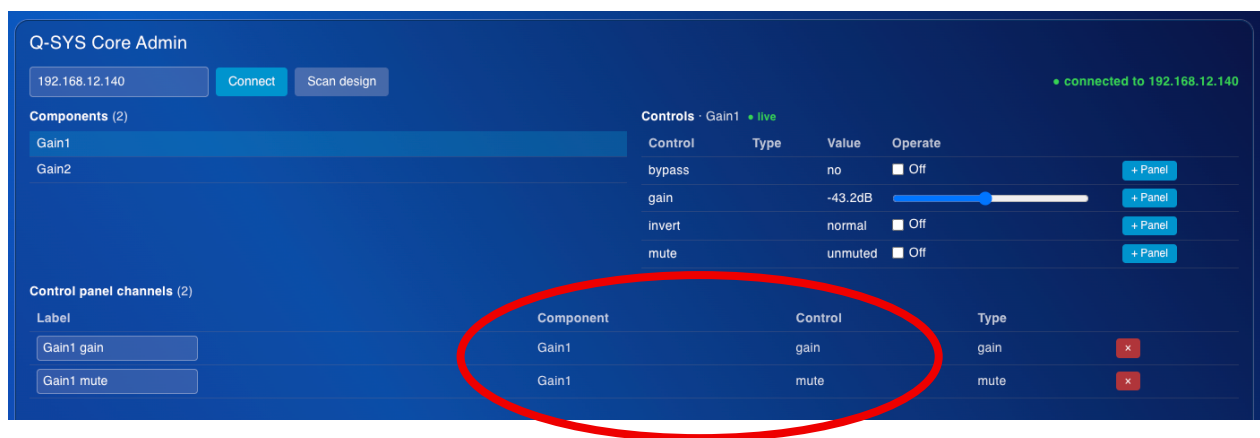
When you select an element in UI Studio, you'll find an ID field in the Properties/General section in the upper right corner.

The following section explains in more detail which ID you need to enter here.

The whole bridge is driven by the widget **ID**. Give a UI Studio widget an ID that encodes its Q-SYS target:

```
qsys__<Component>__<control>[<suffix>]  
qsys__<Component>__<control>_txt      (live text label)
```

Two underscores (__) separate the parts, because component and control names may themselves contain single underscores (e.g. Control1_Router_8x8) or dots (select.1).



You can find the appropriate labels for the desired function in the admin dashboard, as mentioned earlier. In our example, the labels combine to form the following ID:

Example Gain1 Mute:

qsys__<Component>__<control> = qsys__Gain1__mute_on = ID for Button “Mute”

Example Gain1 Unmute:

qsys__<Component>__<control> = qsys__Gain1__mute_off = ID for Button “Unmute”



Two underscores (__) separate the parts

5 ID Syntax

Fader (gain)

A UI slider (0–100) maps to Q-SYS position 0.0–1.0. Add a text with the same id plus `_txt` for the live dB readout.

```
UI Slider:  
qsys__Gain1__gain  
  
Ui Button (set fader value to -6dB):  
qsys__Gain1__gain_set-6  
  
Ui Text (live dB string):  
qsys__Gain1__gain_txt
```

Fader step buttons (+ / -)

A UI button whose id ends in `__up` / `__down` nudges the gain by 1 dB. Append a number for a larger step (`__up3` = +3 dB). Requires the control to be subscribed (reads the current value).

```
UI Buttons (step 1 dB):  
qsys__Gain1__gain__up  
qsys__Gain1__gain__down  
  
UI Buttons (step e.g. 3dB):  
qsys__Gain1__gain__up3  
qsys__Gain1__gain__down3
```

Mute - single toggle button

A UI button with the bare control id toggles the boolean. Feedback drives its state: `state-active` (glows) when muted, `state-inactive` (dimmed) otherwise.

```
UI Buttons:  
qsys__Gain1__mute
```

Mute / Unmute - two buttons

Fixed-value buttons: `__on` sets true, `__off` sets false. The active one is highlighted automatically.

```
UI Buttons:  
qsys__Gain1__mute__on  
qsys__Gain1__mute__off
```

Trigger (snapshot recall, reset, PTZ)

A button ending in `__trigger` fires a momentary 1. A Snapshot component exposes `load.1 ... load.N`; each recall bank is one button.

```
UI Buttons:
qsys__Snapshots1__load.1__trigger
qsys__Snapshots1__load.2__trigger
```

Source / preset select

For a single control that takes an integer index (a router output's `select.1`), use `__set<N>` – one button per choice. The bridge highlights the active source and clears the previous one (radio behaviour).

```
UI Buttons:
qsys__Control_Router_8x8__select.1__set1
qsys__Control_Router_8x8__select.1__set2
qsys__Control_Router_8x8__select.1__set3
```

Level meters (RMS / peak)

Q-SYS exposes signal levels through the Meter component (RMS, Peak, RMS+Peak, True Peak). Add it in Q-SYS Designer, give it a Code Name (e.g. Meter1), set Script Access = External, and feed your audio into it. Each channel is a read-only control - typically `meter.1`, `meter.2` - whose value is the level in dB and whose position is normalised 0.0–1.0. Many blocks (gains, mixer channels) also carry built-in output meters you can read the same way.

No bridge changes are needed - a level is just a control that updates continuously. The feedback path already drives the Interactive UI gauge and bargraph widgets, so a meter maps straight onto a level bar. Read-only: no button, no set.

```
UI Bargraph (segmented bar):
qsys__Meter1__meter.1

UI Gauge (continuous 0..100 bar):
qsys__Meter1__meter.1

UI Text - numeric dB readout:
qsys__Meter1__meter.1_txt
```

Level meters update rate. The engine polls at 0.2 s (5 Hz). This is fine for a level / presence indicator but is not a smooth 30–60 Hz VU meter. Meters also generate constant change-group traffic, so subscribe a few, not dozens, at once. A dedicated faster poll rate for a single meter channel can be added if a smoother bar is required.

6 Suffix reference

All suffixes attach to `qsys__<Component>__<control>`:

Widget / suffix	Press action	Example id
slider (none)	set position 0–100 → 0.0–1.0	<code>qsys__Gain1__gain</code>
button (none)	toggle boolean	<code>qsys__Gain1__mute</code>
<code>__on / __off</code>	fixed true / false	<code>qsys__Gain1__mute__on</code>
<code>__up / __down</code>	step ±1 dB (<code>__up3</code> = 3 dB)	<code>qsys__Gain1__gain__up</code>
<code>__trigger</code>	momentary 1	<code>qsys__Snap1__load.1__trigger</code>
<code>__set<N></code>	set fixed value N (auto-highlight)	<code>qsys__Router1__select.1__set2</code>
<code>__txt</code>	live text / dB string	<code>qsys__Gain1__gain_txt</code>

Reserved end words: a control literally named on/off/up/down/set/trigger would collide with a suffix. Standard Q-SYS names (gain, mute, load.N, select.N) are unaffected.

7 End-to-end examples

Channel strip (Gain1)

Admin: add Gain1/gain and Gain1/mute as channels

UI Slider:	Fader:	<code>qsys__Gain1__gain + dB</code>
UI Button:	Fader step -3dB:	<code>qsys__Gain1__gain__up3</code>
UI Button:	Fader step +3dB:	<code>qsys__Gain1__gain__down3</code>
UI Button:	Fader set 0dB:	<code>qsys__Gain1__gain__set0</code>
UI Button:	Fader set -6dB:	<code>qsys__Gain1__gain__set-6</code>
UI Text:	Fader:	<code>qsys__Gain1__gain_txt</code>

UI Button:	Mute:	<code>qsys__Gain1__mute__on</code>
UI Button:	Unmute:	<code>qsys__Gain1__mute__off</code>
UI Button:	Toggle Mute:	<code>qsys__Gain1__mute</code>

Router source selector (8x8, output 1)

Admin: add Control_Router_8x8/select.1 as a channel

UI Button: Router IN1: qsys__Control_Router_8x8__select.1__set1

UI Button: Router IN2: qsys__Control_Router_8x8__select.1__set2

UI Button: Router IN3: qsys__Control_Router_8x8__select.1__set3

Snapshot recall

UI Buttons: qsys__Snapshots1__load.1__trigger

UI Buttons: qsys__Snapshots1__load.2__trigger

UI Buttons: qsys__Snapshots1__load.3__trigger

Bargraph

UI Bargraph: RMS: qsys__Meter1__meter.1

UI Text: RMS: qsys__Meter1__meter.1_txt